

國立交通大學  
工業工程與管理學系

碩士論文

考慮工件族整備時間之雙流線型工廠排程

Dual Flow Shops Scheduling with Family Setup Times



研究生：林慈盈

指導教授：巫木誠 博士

中華民國九十八年六月

考慮工件族整備時間之雙流線型工廠排程

## Dual Flow Shops Scheduling with Family Setup Times

研究生：林慈盈

Student : Cih-Ying Lin

指導教授：巫木誠 博士

Advisor : Dr. Muh-Cherng Wu

國立交通大學

工業工程與管理學系



Submitted to Department of Industrial Engineering and Management  
College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master of Science

in

Industrial Engineering

June 2009

Hsinchu, Taiwan, Republic of China

中華民國九十八年六月

# 考慮工件族整備時間之雙流線型工廠排程

研究生：林慈盈

指導教授：巫木誠 博士

國立交通大學工業工程與管理學系碩士班

## 中文摘要

本研究探討考慮跨廠與整備時間的雙流線型工廠排程問題。此排程之研究目標為最小化寬裕時間之變異係數( $\text{Min } CV_s$ ,  $CV_s = \frac{\sigma_s}{\bar{X}_s}$ )，寬裕時間即為交期與完工時間之間的差距。於此問題中，我們將需要相同整備治具的工件集合稱之為工件族。大多數以往的文獻中，不是使用 *family-based* (將全部皆為同一家族的工件視為單一工件做排程)，就是使用 *individual-based* (每個工件排程是獨立地，而不考慮加入相同工件族)。本研究提出了 *group-based* 的方法 (亦即將一個工件族區分成多個工件集合，排程中將每個工作集合當作單一個體)。此三個以基因演算法為基礎的 GA-EDD-Family、GA-EDD-Group 及 GA-EDD-Individual 方法被發展以及經由多種實驗做比較。實驗結果指出 GA-EDD-Group 的方法在多數情境下勝過於其他兩種方法。

**關鍵詞：**排程、跨廠、雙流線型生產、工件族、整備時間、交期、基因演算法

# Dual Flow Shops Scheduling with Family Setup Times

Student : Cih-Ying Lin

Advisor : Dr. Muh-Cherng Wu

Department of Industrial Engineering and Management

National Chiao Tung University

## Abstract

This research examines a dual flow shop scheduling problem, which is in the context of considering cross-plant processing and setup times. The scheduling objective is to minimize coefficient of variation of slack times ( $Min CV_s$ ,  $CV_s = \frac{\sigma_s}{\bar{X}_s}$ ), in which the slack time of a job denotes the difference between the due date and its total processing time. Herein, a set of jobs that need the same setup is called a job family. Most prior literature either used a *family-based* approach (all jobs of a particular family are scheduled as a single job) or used an *individual-based* approach (each job is independently scheduled without considering its affiliation to its job family). This research proposes a *group-based* approach (that is, dividing a job family into several job groups, and scheduling each job group as a single entity). Several genetic algorithms (GAs), which are of GA-EDD-Family, GA-EDD-Group, or GA-EDD-Individual, have been developed and compared by numerical experiments. Experiment results indicate that the *group-based* approach outperforms the other two approaches in most scenarios.

**Keywords** : scheduling; cross-plant; dual flow shop; family; setup time; due date; genetic algorithm(GA)

## 誌謝

本論文得以順利完成，首先要感謝巫木誠老師這兩年來細心的教導。除了在學術上獲益良多外，老師時常灌輸我們一些待人處世的道理，給予我們正向的想法與建議，以及強調誠信的重要。很幸運能夠跟著老師學習做事的方法、有紀律的生活以及思考的模式，老師的指導讓學生受用無窮，在此致上萬分的感謝。同時也感謝口試委員許錫美老師、彭德保老師和陳文智老師在論文口試時，給予許多寶貴的意見與指導，讓本論文更加完整。

在兩年的碩士生活中，特別感謝博士班邱志文學長在下班後抽空指導我們程式語言的學習，使我們紮實的學會此工具，並也針對論文提供了寶貴的意見。此外，還要感謝劉謹銘學長提供業界的工廠資訊，讓我們得以順利完成論文。另外，感謝博士班蘇泰盛學長、施昌甫學長、陳振富學長，以及同研究室的同學林昭宏、陳文旻、呂佳玟、黃亮銓這兩年的陪伴，彼此討論研究的問題，互相激勵與安慰，使我的碩士生活充實而歡樂。



最後，我要感謝的是我最親愛的家人，在論文撰寫期間給予我很大關懷與鼓勵，你們的體諒讓我得以專心在論文的研究上，感謝你們這麼多年無怨無悔的付出與辛勞。在此，謹以此論文獻給我最敬愛的家人、師長與朋友。

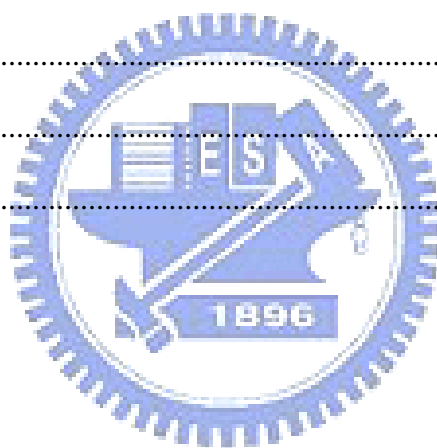
林慈盈 于 新竹交大

2009 年 6 月

# 目 錄

中文摘要.....	i
<b>Abstract.....</b>	<b>ii</b>
誌 謝.....	iii
目 錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	2
1.3 研究議題.....	3
1.4 研究範圍與限制.....	4
1.5 章節概要.....	5
第二章 文獻探討.....	6
2.1 工件族整備時間與交期限制.....	6
2.2 水平式多廠區跨廠生產型態.....	8
2.3 彈性流線型生產與雙流線型生產比較.....	9
2.4 基因演算法 (Genetic Algorithm).....	11
2.4.1 基因演算法執行流程.....	11
2.5 文獻探討結語.....	14
第三章 研究方法.....	15
3.1 研究問題介紹.....	15
3.2 基因演算法.....	16
3.2.1 染色體設計與編碼.....	16
3.2.2 染色體解讀.....	17

1.加工途程解讀.....	17
2.加工順序解讀.....	21
(1)群組演算法(GA-EDD-Group) .....	21
(2)家族式派工(GA-EDD-Family) .....	25
3.2.3 適合度函數.....	27
3.2.4 染色體交配、突變.....	31
3.2.5 染色體複製.....	33
3.2.6 終止條件.....	33
第四章 實驗分析與結果.....	34
4.1 求解品質.....	35
4.1 求解速度.....	38
第五章 結論與未來研究.....	39
參考文獻.....	41



## 表目錄

表 3.1 加工途程-加工時間資訊.....	19
表 3.2 各工件交期及工件族之相關資訊-(1).....	23
表 3.3 各工件交期及工件族之相關資訊-(2).....	26
表 4.1 三種演算法各情境之實驗數據.....	36
表 4.2 家族式派工與群組演算法各情境之改善率.....	37





## 圖目錄

圖 1.1 雙流線型生產(Dual Flow Shop)-(1).....	3
圖 2.1 垂直式多廠區生產型態.....	9
圖 2.2 水平式多廠區生產型態.....	9
圖 2.3 彈性流線型生產(Flexible Flow Shop) .....	10
圖 2.4 雙流線型生產(Dual Flow Shop)-(2).....	10
圖 2.5 基因演算法執行流程.....	12
圖 3.1 雙流線型生產(Dual Flow Shop)-(3).....	15
圖 3.2 染色體設計與編碼.....	16
圖 3.3 三站染色體解讀.....	17
圖 3.4 未做加工途程前的染色體解讀.....	19
圖 3.5 分廠後染色體解讀.....	21
圖 3.6 工件 3 之加工途程.....	21
圖 3.7 染色體經加工途程解讀後的染色體排序-(1).....	23
圖 3.8 EDD 排序後的染色體解讀-(1) .....	23
圖 3.9 J5 往前與 J3 合併之染色體解讀 .....	24
圖 3.10 J9 往前與 J4 合併之染色體解讀 .....	24
圖 3.11 J8 往前與 J1 合併之染色體解讀.....	25
圖 3.12 染色體經加工途程解讀後的染色體排序-(2).....	26
圖 3.13 EDD 排序後的染色體解讀-(2) .....	27
圖 3.14 工件族間的排序.....	27
圖 3.15 家族式派工後的最佳排序.....	27
圖 3.16 上一順位工件之完成時間加上整備時間晚於工件到達時間-(1).....	30
圖 3.17 上一順位工件之完成時間加上整備時間晚於工件到達時間-(2).....	30
圖 3.18 上一順位工件之完成時間加上整備時間早於工件到達時間.....	30

圖 3.19 線性順序交配(LOX).....	32
圖 3.20 突變方式(Swap).....	32
圖 3.21 俄羅斯輪盤法.....	33
圖 4.1 群組演算法之工件族個數不同的改善率趨勢.....	37
圖 4.2 群組演算法之整備時間不同的改善率趨勢.....	38
圖 4.3 三種演算法之求解速度.....	38



# 第一章 緒論

## 1.1 研究背景與動機

由於電子產品科技發展迅速，加上消費者對於電子產品的功能需求越來越多，產品種類不斷推陳出新，為了滿足消費者大量的需求，過去單廠的傳統生產模式，已經不能滿足目前市場的消費型態。企業為了滿足目前市場的消費型態，藉由擴大工廠產能以承接大量訂單，促使企業朝向整合上下游供應鏈的垂直式擴廠生產模式或另外建造相同生產工廠的水平式擴廠生產模式，使得工廠由傳統的單廠生產模式逐漸擴大為多廠區生產模式。

本研究的背景則是針對建造相同生產工廠的水平式多廠區之生產型態加以討論，水平式多廠區生產模式又分為跨廠與不跨廠兩種生產型態，跨廠生產是指工件可以選擇跨廠區進行不同加工製程，不跨廠則是指工件從頭到尾的製程皆要在同一廠中完成加工，前者的特色為必需決定跨廠生產途程之產能分配及生產排程之決策；後者則是單一企業統一接受訂單後，評估各廠廠能做訂單的分配，決定訂單分配後，各廠則各自排程生產，如此分成兩階段決策的不跨廠生產，容易造成產能無法充分利用。然而，本研究為了使產能可以充分利用，增加了跨廠支援模式。

本研究亦考慮工件族整備時間的製程限制，在以往的研究中，整備時間通常選擇被忽略或包含於加工時間中，但在實際工廠中，整備時間是影響生產績效不可忽略的因素，尤其在考慮有工件族之工件加工的情況下，整備時間就顯的格外重要(工件族即為有相似製程需求的工件，如：使用相同治具)，由於當相同工件族的工件連續加工，可省略更換治具的動作，以節省整備時間，相對可提升產能利用率，增加產出效率。

然而，雖然將相同工件族之工件排在一起即可省略整備時間，但是因為各工件皆有不同的交期限制，若強制將相同工件族工件全部合併，雖然可以節省大量的整備時間，

但是卻沒有考慮到交期限制，容易造成交期晚的工件為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，導致工件遲交。所以如何兼顧節省整備時間及交期的限制，使兩者之間取得平衡，便是本研究的重點。

綜合以上，當工廠由單廠擴大為多廠區生產模式，單廠的傳統生產排程可能會不適用於複雜的多廠區，本研究針對水平式跨廠區、流程型(Flow Shop)的多廠生產模式，並考量有工件族情況下的整備時間以及交期限制，整體規劃穩健的多廠區跨廠生產排程，讓產能充分發揮。因此如何運用跨廠區生產排程滿足顧客原本已下之訂單提前交貨(Pull In)的需求，以減少重新排程頻率，增加企業運作的效率，是本研究的研究動機。

## 1.2 研究目的

企業的規模逐漸擴大，工廠生產環境由單一工廠日漸擴展到多廠區生產模式，在多廠區生產模式下，為因應顧客原本已下之訂單提前交貨(Pull in)之經常性需求，又得確保原排程效益，因此企業期望有足夠的寬裕時間滿足顧客提前交貨的需求，以減少重新排程的頻率，並獲得顧客的信賴。由於我們不能預期哪個顧客會要求提前交貨，因此期望各訂單在交貨前的寬裕時間變異( $\sigma_s$ )小，如此即能使各訂單允許提前交貨的寬裕一致。同時，又期望能使寬裕時間平均( $\bar{X}_s$ )大，如此可允許訂單提前交貨的寬裕較大，因此為了同時可以考量到交貨前寬裕時間的平均及變異，訂定最小化寬裕時間之變異係數( $Min CV_s, CV_s = \frac{\sigma_s}{\bar{X}_s}$ )為本研究的績效指標。又由於此目標值為望小，所以必須限制寬裕時間平均為正數( $\bar{X}_s \geq 0$ )，否則目標值雖然很小，但寬裕時間平均卻也很小( $\bar{X}_s \leq 0$ )，違反了本研究的目的。

然而，當企業承接大量訂單後，如何善用既有產能，做整體訂單分配且決定生產順序，以充分發揮產能，減少產能閒置，同時在交貨日(Due Date)多變環境並考慮有工件

族的整備時間情況下，建立較穩健的生產排程，故本研究希望求得近似最佳績效之下能夠：

1. 決定各工件跨廠加工途程(Route Assignment)。
2. 決定各機台工件加工順序(Job Sequencing)。

### 1.3 研究議題

本研究探討的工廠為雙流線型工廠，此雙流線型工廠共有 A、B 兩廠，每廠各有三道加工流程，一站各有一部加工機台，兩廠同一站有相同的機台，可做跨廠支援加工(如圖 1.1)。所有工件都必須依序經過此三道加工流程才算完成加工，各工件在每站皆可選擇 A、B 兩廠做跨廠加工，故每個工件皆有八種加工途程可作選擇( $2^3=8$ )，其中包含兩種不跨廠加工途程( $S_{A1} \rightarrow S_{A2} \rightarrow S_{A3}$ ； $S_{B1} \rightarrow S_{B2} \rightarrow S_{B3}$ )及六種跨廠加工途程( $S_{A1} \rightarrow S_{A2} \rightarrow S_{B3}$ ； $S_{A1} \rightarrow S_{B2} \rightarrow S_{A3}$ ； $S_{A1} \rightarrow S_{B2} \rightarrow S_{B3}$ ； $S_{B1} \rightarrow S_{B2} \rightarrow S_{A3}$ ； $S_{B1} \rightarrow S_{A2} \rightarrow S_{B3}$ ； $S_{B1} \rightarrow S_{A2} \rightarrow S_{A3}$ )。在此工廠內我們希望能夠求得近似最佳績效下，決定各工件的加工途程以及各機台的工件加工順序。

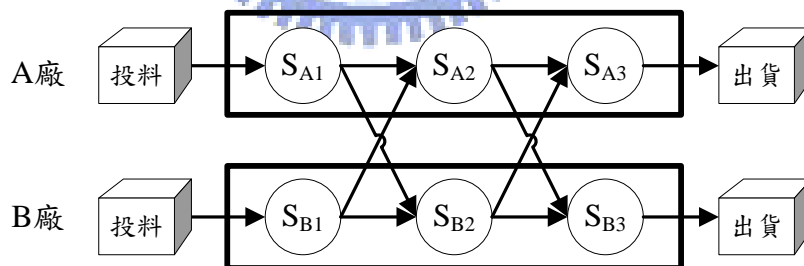


圖 1.1 雙流線型生產(Dual Flow Shop)-(1)

本研究並考慮工件族之整備時間，所有工件所歸屬之工件族皆為已知，而各工件所歸屬的工件族不會因為加工站別不同而跟著改變。工件族之整備時間主要的特性是當相同工件族之工件排序一起加工時，只需要一次的整備時間即可。如此將相同工件族的工件排序在一起加工能夠節省整備時間，那麼強制將所有相同工件族的工件全部排序一起加工，即可節省大量的整備時間，但卻沒有考慮到交期的限制，容易造成交期晚的工件

為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，反而讓績效指標變差，並不適合本研究的研究目標。故本研究在雙流線型工廠下，如何兼顧節省整備時間及交期的限制，使兩者之間取得平衡，便是本研究的重要議題。

#### 1.4 研究範圍與限制

本研究題目為「考慮整備時間之雙流線型工廠跨廠排程」，意即一企業內有兩座工廠，此兩個工廠生產型態皆為流線型工廠，兩廠間同一站有相同加工製程機台，可允許跨廠支援加工，並考慮在有工件族整備時間之雙廠跨廠生產途程指派及生產順序之決定，此系統較為複雜，為了縮小研究的範圍，在不影響真實性的前提下，訂出以下的限制與假設：

廠限制與假設：

1. 本研究探討的對象為雙流線型工廠
2. 屬接單式生產環境(Make to Order；MTO)
3. 起始階段為空廠狀態(無在製品)
4. 時間開始加工時所有工件已準備就緒
5. 不考慮拆單併單的情形
6. 不考慮外包
7. 不考慮機台當機
8. 不考慮成本問題
9. 不考慮回流製程
10. 不考慮良率問題

跨廠限制與假設：

1. 兩廠相同的加工製程可作跨廠支援加工。
2. 兩廠加工效率一樣，且工件在各廠各站的加工時間為已知。
3. 需考慮跨廠運輸時間且為已知。

工件限制與假設：

1. 一張訂單同等於一張工單，工單縮小為一個工件來看，而每張工單做相同的產品，工單批量不可被切割。
2. 每個工件皆須經過相同的加工製程。
3. 每個工件都需有各自的交貨日期(Due Date)。

工件族限制與假設：

1. 各工件所歸屬之工件族為已知。
2. 各站皆考慮工件族整備時間，且在各廠各站之整備時間相同且為已知。
3. 各工件所歸屬之工件族不會因為站別不同而改變。
4. 整備時間為順序獨立的工件族整備時間(Sequence-Independent Family Setup Time)。



## 1.5 章節概要

本研究論文內容共含五個章節，第一章說明考慮工件族整備時間之雙流線型工廠排程研究的背景與動機、研究目的、研究議題及研究範圍。第二章回顧國內外有關本研究之生產規劃相關文獻作為解決本研究問題之參考。第三章將詳細介紹雙流線型工廠的生產情境，並發展演算法，利用基因演算法求得近似最佳解，以決定跨廠之加工途程及加工順序之決策。第四章為實驗分析與結果，以實際資料，利用 Visual C++6.0 模擬雙流線型工廠運作，驗證研究方法之績效。最後，本研究的結論與未來研究則於第五章中說明。



## 第二章 文獻探討

本研究是探討考慮工件族整備時間的雙流線型工廠，規劃出穩健的生產排程，本章首先針對本研究的重點工件族整備時間與交期限制之間的關係做分析及文獻探討；接著針對水平式多廠區生產型態做文獻回顧；然後再針對彈性流線型生產(Flexible Flow Shop)與雙流線型生產(Dual Flow Shop)做生產型態的比較；最後，由於求解空間大，若利用整數規劃求解時間長，因此本研究求解工具使用基因演算法來快速求得近似最佳解，亦再針對基因演算法做文獻回顧。

### 2.1 工件族整備時間與交期限制

本研究為在雙流線型生產工廠型態下，必須考慮有工件族情況下的整備時間。整備時間通常發生在兩相鄰工件間，下一個工件加工前先進行更換治具、對準作業或是清理模具的時候。然而，在以往的研究中，常常為了簡化問題的複雜度，將整備時間選擇忽略或包含於加工時間中，但在實際工廠中，整備時間是影響生產績效不可忽略的因素。本研究的整備時間為尚有考慮到工件族的整備時間，我們將相似的工件歸類為同一種產品型態，稱之為工件族，而同一種工件族的工件，通常需要相同的治具整備。Potts et al. (2000)由於當相同工件族的工件連續加工，需要的整備治具相同，所以不需要再做清理或更換的動作，只要在連續加工相同工件族的第一個工件做一次整備時間即可，可省略更換治具的動作。換句話說，越多相同工件族的工件連續加工，即可節省越多整備時間。工件族整備時間的重點是如何在生產排程上作適當的安排，以降低整備時間提升產能利用率、增加產出效率，故在考慮有工件族之工件加工的情況下，整備時間就顯的格外重要。

本研究亦考慮有交期的情況下做雙流線型工廠的排程，由於顧客所下的訂單各有不同的交期限制，我們必須滿足顧客的交期，但又希望同時能符合上段所述，節省工件族整備時間，以增加產出效率。若只偏重節省工件族整備時間，而強制將所有相同工件族



的工件全部排序一起加工，雖然可節省大量的整備時間，但卻沒有考慮到交期的限制，容易造成交期晚的工件為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，反而讓績效指標變差。反之，若只偏重交期限制，全部工件都以EDD(Early Due Date)排序那麼就損失了節省工件族整備時間所能達到的效益。綜合以上所述，我們可以發現節省工件族整備時間以及交期限制之間是一種權衡關係(Trade-off)，故本研究在雙流線型工廠下，如何兼顧節省整備時間及交期的限制，使兩者之間取得平衡，便是本研究的重要議題。以下則是針對同時存在工件族整備時間與交期限制之考量的文獻探討。

首先Yuan et al. (2006)提出在單機批量排程問題中，目標最小化總完工時間(Makespan)，以及Jolai (2005)目標最小化延誤工件數(Minimizing Number of Tardy Jobs)下，證明在有工件族整備時間及交期限制下，為Np-hard的問題。

Yimer et al. (2009)提出利用混整數模糊規劃模式(Mixed-integer Fuzzy Programming Model)及GA解決二階流程型(Two-Stage Flowshop)排程問題，是將各獨立的工件分類成好幾個批量(Batches)，相同批量中有相同的工件族，排程決策為必須決定批量間的排序及批量裡工件的序。而Srinoi et al. (2006)，亦利用模糊邏輯(Fuzzy Logic)解決彈性製造系統(Flexible Manufacturing Systems, FMS)之排程問題。

Pearn et al. (2008)在有平行機台及製程迴流(Reentry)的工廠中，提出兩個排程決策(1)決定各工件在每站的交期，以避免有交期的問題發生在最末站(2)決定各站工件的排序。Omar et al. (2006)考慮等效平行機台的排程問題，利用混整數規劃模式(Mixed Integer Programming Formulation Model)可獲得最佳解，最多可解決18個工件及4個工件族的問題。

Eom et al. (2002)考慮平行機台的排程問題，最小化加權延誤時間(Minimize the Total Weighted Tardiness)三階段啟發式演算，分別為：(1)先利用EDD排序，再將工件依

據交期的相似性，分類成小工件集合(Small Job-sets)；(2)使用Tabu搜尋法，排序各工件組內的工件；(3)決定各工件的機台配置。

然而，本研究所提出的群組演算法(GA-EDD-Group)，不同於以往針對工件族整備時間與交期限制同時存在的研究，目的為同時兼顧節省工件族整備時間以及交期限制，以解決此種權衡問題，我們將在第三章做詳述。

另外，針對工件族整備時間的分類，Potts et al. (2000)將工件族整備時間分為兩種類型：

1. 順序獨立之群組整備時間(Sequence-Independent Family Setup Time)：整備時間不會因群組順序不同而異，亦即處理各工件族前所需的整備時間長短與前一個被處理的工件族無關。本研究之工件族整備時間歸類為順序相依之群組整備時間。
2. 順序相依之群組整備時間(Sequence-Dependent Family Setup Time)：整備時間因群組順序不同而異，亦即處理各工件族前所需的整備時間長短會隨著前一個被處理的工件族而改變。

本研究為考慮順序獨立之群組整備時間之研究。

## 2.2 水平式多廠區跨廠生產型態

在考量工件族整備時間與交期的關係後，我們將此考量運用在多廠區，且增加跨廠支援模式的生產型態工廠，此小節將針對多廠區生產型態做分類。針對多廠區的型態的文獻，各有不同的定義，大致上我們將多廠區定義分為兩類，若多廠區之工廠型態為像供應鏈般的生產型態(如：TFT-LCD 三階段製程，每一階段分別為一個廠區)，每廠區的製程各有不同，其多廠區之生產規劃型態即為垂直式多廠區生產型態(如圖 2.1)，相關研究有 Moon et al. (2002)、Guinet (2001)、Lee et al. (2002)、Gnoni et al. (2003)、Levis et al. (2004)、Moon et al. (2005)、Bonfill et al. (2008)、李佳諭 (2006)。然而，

多廠區的生產型態若為具有多座類似的工廠，每廠區的製程是相同，即為水平式多廠區生產型態(如圖 2. 2)，Wu et al.(2008)為研究半導體水平式雙廠區產能相互支援的途程規劃，首先利用線性規劃數學式，求取最佳的各產品加工途程組合，接著評估切割點與生產比例組合的績效，最後利用基因演算法求取最佳的途程生產比例，在目標流程時間下最大化整體產出。而 Vercellis (1999) 、 Toba (2005) 、 Sambasivan et al.(2005) 、 Wu & Chang (2007)亦為探討水平式多廠區生產型態之相關研究，其中 Toba (2005) 、 Wu & Chang (2007) 、 Wu et al.(2008)為增加考量跨廠相互支援的水平式多廠區之生產型態，以增加產能利用率。本研究的工廠型態為具有多座生產功能類似的工廠，且亦為跨廠支援的模式，故應被歸類為水平式多廠區之跨廠的生產型態，本研究則將此生產型態命名為雙流線型工廠(Dual Flow Shop)。

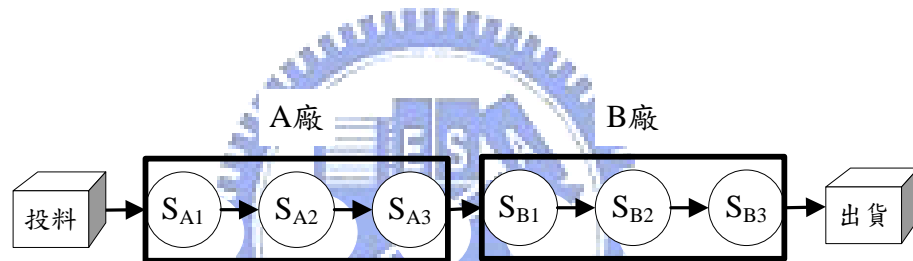


圖 2.1 垂直式多廠區生產型態

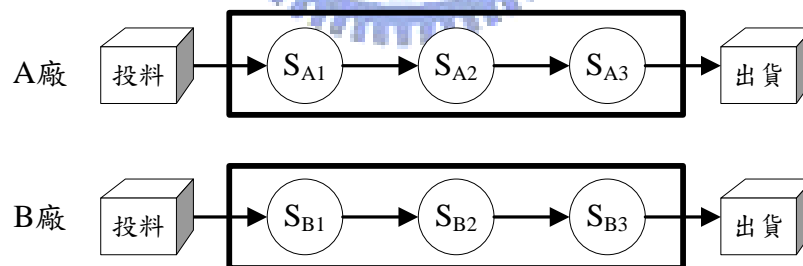


圖 2.2 水平式多廠區生產型態

### 2.3 彈性流線型生產與雙流線型生產比較

根據上節所提到的雙流線型工廠(Dual Flow Shop)，與彈性型流線型工廠(Flexible Flow Shop)有些類似，但其實他們的生產型態是不同的，本小節則針對此兩種生產型態做比較。彈性流線型生產如圖2.3所示，其中 $S_{ij}$ 代表第 $i$ 廠的第 $j$ 站，單廠各站皆有功能相似的平行機台( $S_{A1}$ 、 $S_{A1'}$ 與 $S_{A2}$ 、 $S_{A2'}$ 與 $S_{A3}$ 、 $S_{A3'}$ 分別為平行機台)，Allahverdi et al. (2008)

提到在彈性流線型生產環境，每個工件必須依序經過第一站、第二站…等製程。彈性流線型生產是指單廠中每一站皆有功能相似的平行機台可互相彈性支援利用，只考量單站的產能來決定加工途程，意即單站機台產能不足，只考慮同站之平行機台做加工，並沒有考慮整廠之產能。此外，彈性流線型生產沒有考量跨廠運輸時間，而廠內搬運時間通常被忽略或包含於加工時間內，由於彈性流線型生產沒有跨廠運輸時間，故工件在上站的途程選擇不會因為跨機台而影響下站的途程選擇。

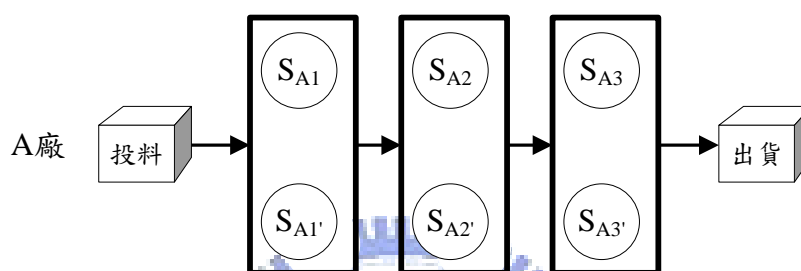


圖 2.3 彈性流線型生產(Flexible Flow Shop)

雙流線型生產如圖 2.4 所示，其中  $S_{ij}$  代表第  $i$  廠的第  $j$  站，兩廠各站有功能相似的平行機台 ( $S_{A1}$ 、 $S_{B1}$  與  $S_{A2}$ 、 $S_{B2}$  與  $S_{A3}$ 、 $S_{B3}$  分別為平行機台)，與彈性流線型生產環境相同的地方為每個工件必須依序經過第一站、第二站…等製程。雙流線型生產是指兩廠中每一站皆有功能相似的平行機台可做跨廠支援加工，必須考量兩廠整體的產能來決定加工途程。此外，雙流線型生產考量跨廠運輸時間，由於運輸時間長，會直接影響到產出績效，也會影響是否跨廠的決策，因此無法將運輸時間併入加工時間或忽略。由於考量到運輸時間的重要性，故工件在途程選擇上會因為跨廠運輸時間而影響，進而影響產出績效。

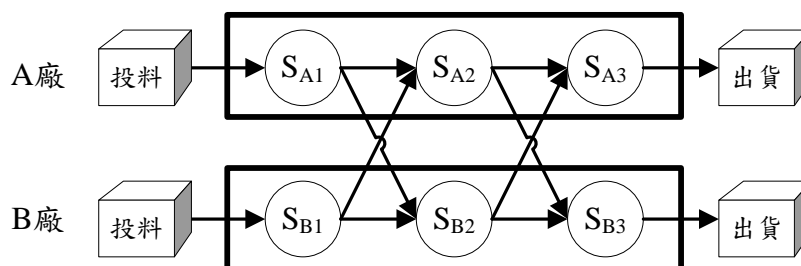


圖 2.4 雙流線型生產(Dual Flow Shop)-(2)

綜合以上，雙流線型生產之運輸時間會影響工件的加工途程，必須考量兩廠整體的產能，因此工件上站的途程選擇會影響下一站的途程選擇；然而，由於彈性流線型生產的廠內搬運時間通常被忽略或包含於加工時間內，只考量到單站的機台產能，只選擇單站中較有空的機台進行加工，並沒有考量到整體產能，故工件上站的途程選擇並不會影響下一站的途程選擇。因此，雙流線型生產與彈性流線型生產排程並不相同，而在過去研究中亦沒有找到與雙流線型生產相似之文獻。

## 2.4 基因演算法 (Genetic Algorithm)

基因演算法是由 Holland (1975)提出，為模仿生物演化特性的「自然遺傳」系統，經由競爭、淘汰，唯有適應力強的物種得以生存及繁衍，並透過複製(Reproduction)、交配(Crossover)、突變(Mutation)等演化方式產生下一代物種。

### 2.4.1 基因演算法執行流程

基因演算法執行之前，必須定義適合度函數值(Fitness function)的計算方式，並設定好各項參數值，如：GA 停止條件、母體大小(Population Size)、交配率(Crossover Rate)、突變率(Mutation Rate)……等。接著設計染色體的編碼(Gene Encoding)方式，並隨機產生初始母代(Initial Population)，即染色體數量達成母體大小為止，接著對母代的各個染色體進行解碼且計算各染色體的適合度函數值，並選擇此初始母代的最佳適合度函數值為全域最佳解( $\pi^*$ )，接著透過母代的兩兩染色體互相交配 (Crossover) 及單一染色體突變 (Mutation) 而成為子代，並個別計算出子代的染色體適合度函數值，混合母代與子代的染色體形成一個池 (Pool)，並選擇池中最佳適合度函數值為目前最佳解( $\pi^0$ )與全域最佳解比較( $\pi^*$ )，若目前最佳解( $\pi^0$ )較好則取代全域最佳解( $\pi^* = \pi^0$ )，反之，則不取代。然後依據池中染色體的適合度函數值進行篩選複製，適合度函數值較佳的染色體有較大的機率會被複製，而複製之染色體數量則達到母體大小為止，此即為新一代的母代，如此反覆進行，演化直到達成先前設定的 GA 停止條件才結束。如此一代一代進化過程，

最後保存適合度函數值最好的染色體。而整個基因演算法的執行流程如圖 2.5 所示。

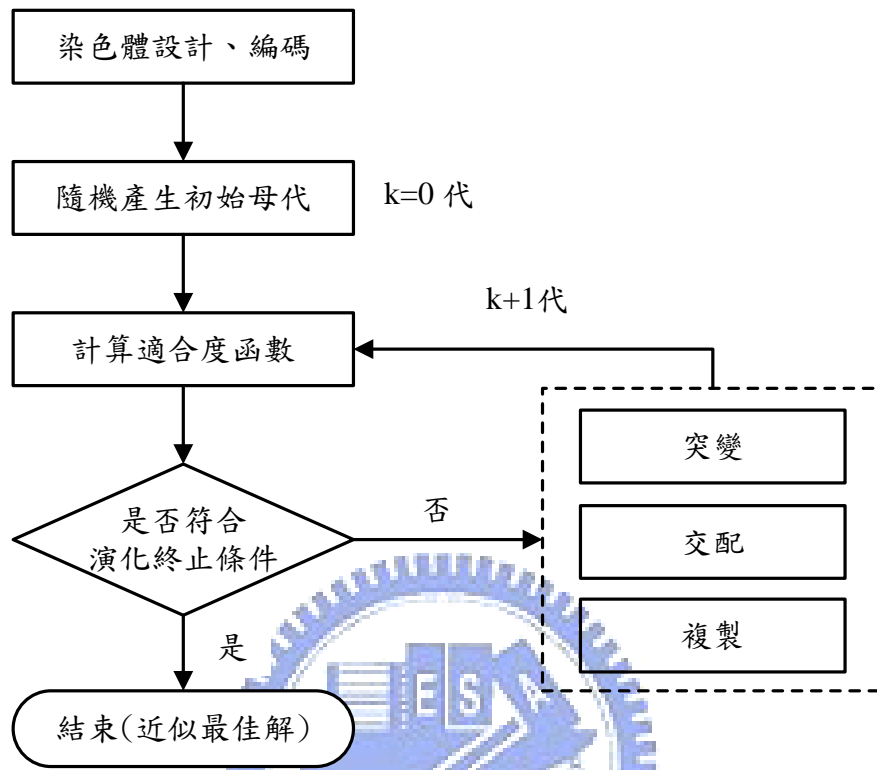


圖 2.5 基因演算法執行流程

以下為依據圖 2.5 基因演算法執行流程作各功能之簡介：

### 1. 染色體設計、編碼(Encoding)

染色體的型態是由一串數字串接而成的字串，而每一個染色體都對應到目標問題的一個解，一般來說染色體越長求解速度越慢。其中染色體的每格基因代表一個字元，將字元編碼並將字串串接起來就組成染色體，染色體編碼亦即將決策變數轉成可以代表可行解的字串。

### 2. 初始母代(Initial Population)

必須先隨機產生  $q$  個染色體，稱之為初始母代。基因演算法的特色為同時可由多個解為出發點開始搜尋，故在執行基因演算法之前，必須設定母體大小(Population Size)，並產生一組可行解之集合，稱之為初始母代。而群體之數目多寡則會直接影響求解品質



及效率，亦即群體數目若太小，不容易廣泛尋找可行解且容易收斂，所求出來的解品質可能會比較差；而若群體數目太大，則必須花費的運算時間就越長。

### 3. 適合度函數值(Fitness function)

母代產生後，必須要有一個衡量基準來評估染色體的好壞，在計算適合度函數之前就必須定義好適合度函數的計算方式，接著將染色體帶入此適合度函數計算，其值越好代表染色體適應力較佳，因此存活的機率也越高，反之較差的染色體則代表染色體適應力較差，而存活的機率則相對較低。

### 4. 交配(Crossover)

由母代中隨機挑選兩個染色體，彼此交換位元資訊進而產生兩個新的個體，原理為藉由累積前代較為優良的位元資訊，以期待能夠產生出更優秀的個體，而事先必須設定交配機率(Crossover Rate)來決定進行交配的數量。



### 5. 突變(Mutation)

演化方式除了交配以外還需搭配突變的使用，突變可以產生與母體變異較大的染色體，目的為防止過早收斂而陷入局部最佳化(Local Optimum)。突變機率的設定通常不大，若過大會影響到解收斂的速度，反之，若突變率過小，則難以產生與母體變異較大的解，不容易跳脫局部解。一般常用的突變有交換(Swap)、插入(Insert)及反轉(Inverse)。

### 6. 複製(Reproduction)

當交配與突變後，交配池中總共有母代再加上子代的染色體，交配池中的染色體依據適合度優劣，來決定下一代存活的機率，適合度函數值較佳的染色體被複製的機率較高，保留到下一代存活，而適合度函數值較差的染色體可能會在此複製階段遭到淘汰。選擇複製方式常見的有兩種：俄羅斯輪盤法(Roulette Wheel Selection)、競賽法(Tournament Selection)。

## 7.演化終止條件

最後，我們必須設定停止條件，使基因演算法演化循環結束，當基因演算法滿足所設定的停止條件時，將終止演化循環，常見的停止條件如下：

1. 達到所設定的最大演化代數上限
2. 達到所設定的最佳解維持代數
3. 演化超過一千個世代後，最佳適合度函數值增加率未超過 1%
4. 達到預先設定的目標函數值

## 2.5 文獻探討結語

綜合以上的文獻探討，可發現雙流線型工廠為尚未充分研究的領域，本研究的排程重點為工件族整備時間與交期限制之間的權衡關係，所發展的群組演算法 (GA-EDD-Group) 又不同以往的研究方法，而整備時間被歸類為以群組為主的順序獨立群組整備時間；然而本研究之雙流線型工廠為水平式多廠區跨廠之生產型態，亦不同於彈性流線型工廠之生產。綜合以上的獨特性，我們可以發現此問題是值得被研究的問題。

本研究由於求解空間大，若利用整數規劃求解，必須要很長的求解時間，因此求解工具使用基因演算法來求解此雙流線型工廠之問題，我們將以基因演算法為架構，於第三章介紹雙流線型工廠排程之研究方法。



### 第三章 研究方法

本章主要目的是在建立雙流線型工廠問題模式並使用基因演算法求解，首先先介紹本研究之問題，接著是以基因演算法為架構介紹研究方法，包含：染色體設計與編碼、染色體解讀、適合度函數及交配、突變、複製的方法及終止條件的設定。其中染色體解讀包含加工途程及加工順序的解讀，而加工順序解讀將介紹本研究所發展的群組演算法(GA-EDD-Group)以及既有的家族式派工(GA-EDD-Family)運用在雙流線型工廠的排程，並將於第四章做實驗比較此兩種方法的優劣。

#### 3.1 研究問題介紹

本研究探討的生產雙流線型工廠，共有 A、B 兩廠，每廠各有三道加工流程，分別為第一站、第二站及第三站，一站各有一部加工機台，兩廠同一站有功能相同的機台，可做跨廠支援加工(圖 3. 1)。現有  $n$  個工件  $J = \{J_1, J_2, \dots, J_n\}$  需加工，每個工件都有各自的交期，所有工件都必須依序經過此三道加工流程(第一站→第二站→第三站)才算完成加工，各工件在每站皆可選擇 A、B 兩廠做跨廠加工，但必須考慮運輸時間。

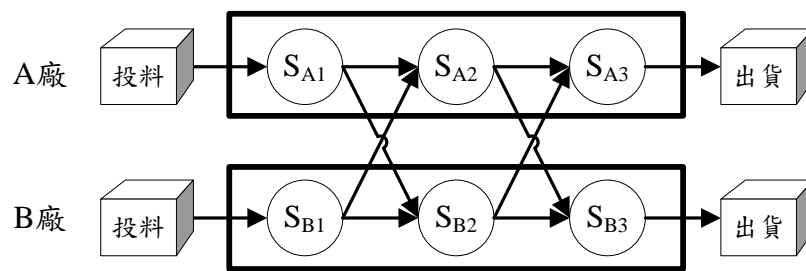


圖 3. 1 雙流線型生產(Dual Flow Shop)-(3)

本研究並考慮工件族之整備時間，所有工件所歸屬之工件族皆為已知，而各工件所歸屬的工件族不會因為加工站別不同而跟著改變。工件族之整備時間主要的特性為各工件在各站加工之前皆必須考慮工件族整備時間，且當相同工件族之工件排序一起加工時則只需要一次的整備時間即可。如此將相同工件族的工件排序在一起加工能夠節省整備

時間，那麼是否本研究就可以利用家族式派工法(GA-EDD-Family)，亦即強制將相同工件族的工件排序一起加工，即可節省大量的整備時間？我們必須考慮到本研究的目標，這種家族式派工法只考慮到節省大量的整備時間，但卻沒有考慮到交期的限制而強制將相同工件族排在一起加工，雖然可以節省很多整備時間，但卻容易造成交期晚的工件為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，反而讓績效指標變差，並不適合本研究的研究目標。

本研究利用工件族可節省整備時間的特性並考慮到各工件有不同的交期限制，發展出群組(GA-EDD-Group)演算法，同時可以節省整備時間及考慮交期的限制，在目標最小化寬裕時間之變異係數( $Min CV_s, CV_s = \frac{\sigma_s}{\bar{X}_s}$ )之下，求解最適加工途程與加工排序。



### 3.2 基因演算法

此雙流線型工廠之問題，由於求解空間很大，導致求解時間長，因此本研究利用基因演算法來快速求解問題，3.2 小節即以基因演算法為架構來介紹研究方法。

#### 3.2.1 染色體設計與編碼

一條染色體代表所需加工的工件數(Job number)，其染色體中的工件不能重複，而每個工件皆歸屬到一個工件族，如圖 3.2 所示，一條染色體中有十格基因格，代表有十個工件需被加工，此十個工件為隨機排序，染色體下方代表的是每個工件的工件族屬性，如：工件 9(J9)所歸屬的工件族為工件族 2 (f2)；工件 5(J5)所歸屬的工件族則為工件族 1 (f1)。

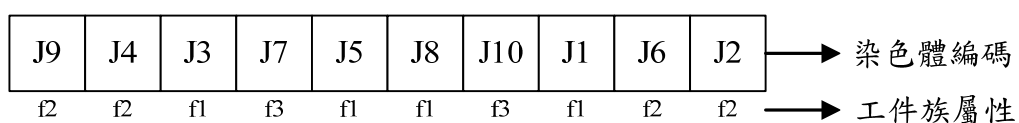


圖 3.2 染色體設計與編碼

經由染色體設計與編碼後，我們必須透過染色體解讀來描述成雙流線型工廠之排程，而本研究透過啟發式演算法(Heuristic Algorithm)來解讀雙流線型工廠之排程，其中包含加工途程解讀及加工順序解讀。故在基因演算法中每產生一條新的染色體，就必須經由染色體解讀成雙流線型工廠之排程後才能做適合度函數的運算，我們將在 3.2.2 節做一連串的介绍

### 3.2.2 染色體解讀

本研究所研究的問題為雙廠跨廠之生產規劃排程問題，每廠各有三站，而如圖 3.3 所示的一條精簡的染色體，要如何代表此研究的問題？經由染色體設計與編碼後，我們將此條精簡的染色體解讀成三條染色體，如圖 3.3 所示，此三條染色體分別代表第一站、第二站及第三站，涵義則為每個工件都必須依序經過此三站做加工。



圖 3.3 三站染色體解讀

接下來，以下小節將分成兩部份解讀，一部份為加工途程，另一部份則為加工順序。

#### 1. 加工途程解讀

經由染色體作三站的解讀後，尚未知每個工件在哪一廠加工，以下則是為了平衡兩廠產能負荷，而做的途程指派(Route Assignment)，亦即指派每個工件在每一站的哪一廠加工，且能使得兩廠的廠能負荷能夠平衡。以下則是本研究為了平衡兩廠產能負荷所做的途程指派。

符號解釋：

$n$ ：工件數

$p$ ：第  $p$  廠， $p = A, B$

$m$ ：第  $m$  站， $m=1, 2, 3$

$(j)$ ：第  $j$  個順位之工件

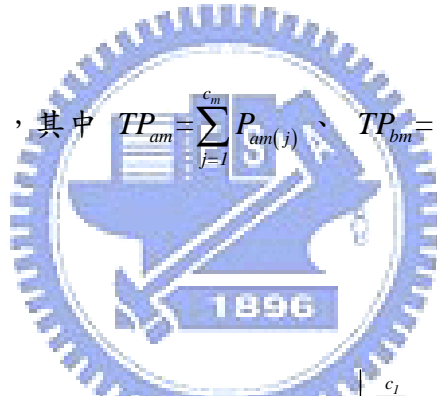
$c_m$ ：第  $m$  站的第  $c$  個順位工件之後為廠分割點位置，為求解變數

$P_{pm(j)}$ ：第  $p$  廠的第  $m$  站的第  $j$  個順位之工件加工時間，為已知

$TP_{pm}$ ：第  $p$  廠的第  $m$  站的總加工時間

分廠公式：

$$\text{Min} \left| \sum_{m=1}^M TP_{am} - \sum_{m=1}^M TP_{bm} \right|, \text{ 其中 } TP_{am} = \sum_{j=1}^{c_m} P_{am(j)}, TP_{bm} = \sum_{j=c_m+1}^n P_{bm(j)} \quad (1)$$



求解步驟：

步驟一： $M=1$  代入公式(1)，得到  $\text{Min} |TP_{a1} - TP_{b1}| = \text{Min} \left| \sum_{j=1}^{c_1} P_{am(j)} - \sum_{j=c_1+1}^n P_{bm(j)} \right|$ ，可求出第一

站的分割點  $c_1$ ，即為第 1 站的第  $c$  個順位工件之後為廠分割點位置。

步驟二： $M=2$  代入公式(1)，得到  $\text{Min} \left| \sum_{m=1}^2 TP_{am} - \sum_{m=1}^2 TP_{bm} \right| = \text{Min} |(TP_{a1} + TP_{a2}) - (TP_{b1} + TP_{b2})| =$

$$\text{Min} \left| \left( TP_{a1} + \sum_{j=1}^{c_2} P_{am(j)} \right) - \left( TP_{b1} + \sum_{j=c_2+1}^n P_{bm(j)} \right) \right|, \text{ 其中 } TP_{a1}, TP_{b1} \text{ 由步驟一已得知，此步}$$

驟可求出第二站的分割點  $c_2$ ，即為第 2 站的第  $c$  個順位工件之後為廠分割點位置。

步驟三： $M=3$  代入公式(1)，得到  $\text{Min} \left| \sum_{m=1}^3 TP_{am} - \sum_{m=1}^3 TP_{bm} \right| = \text{Min} |(TP_{a1} + TP_{a2} + TP_{a3}) - (TP_{b1} + TP_{b2} + TP_{b3})| =$

$$\text{Min} \left( TP_{a1} + TP_{a2} + \sum_{j=1}^{c_3} P_{am(j)} \right) - \left( TP_{b1} + TP_{b2} + \sum_{j=c_3+1}^n P_{bm(j)} \right), \text{ 其中 } TP_{a1}, TP_{a2}, TP_{b1}, TP_{b2}$$

由步驟一、二已得知，此步驟可求出第三站的分割點  $c_3$ ，即為第 3 站的第  $c$  個順位工件之後為廠分割點位置。

釋例：

假設有一條隨機亂數產生的染色體，工件數為 5，在解讀成三條染色體後(如圖 3.4)，為未做加工途程前的染色體解讀，其加工時間資訊如表 3.1。

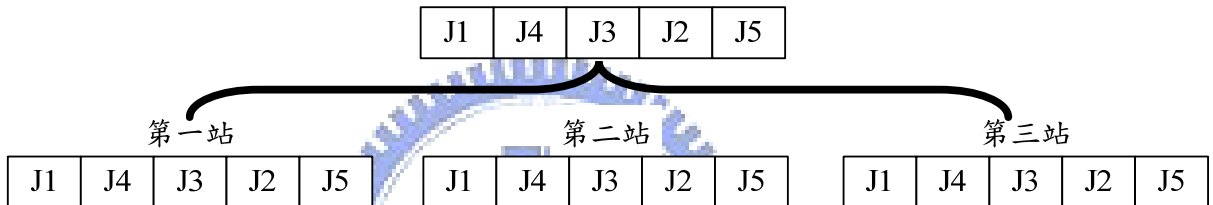


圖 3.4 未做加工途程前的染色體解讀

表 3.1 加工途程-加工時間資訊

站 / 廠 \ 工件		J1	J2	J3	J4	J5
		J1	J2	J3	J4	J5
第一站	A 廠	5	2	3	3	4
	B 廠	4	3	2	4	3
第二站	A 廠	4	2	2	3	4
	B 廠	3	3	3	4	5
第三站	A 廠	5	2	3	3	4
	B 廠	4	3	2	4	3

單位：小時

步驟一： $M=1$  代入公式(1)，得到  $Min|TP_{a1}-TP_{b1}|=Min\left|\sum_{j=1}^{c_1} P_{am(j)}-\sum_{j=c_1+1}^n P_{bm(j)}\right|$ ，求出第一站的

分割點  $c_1=2$ ，可得到最小值  $=|(5+3)-(2+3+3)|=0$ ，因此可以得知在第一站的廠分割點為第二個順位工件之後，亦即第二個順位之前的工件(J1、J4)在 A 廠加工，其餘的工件則在 B 廠加工(J3、J2、J5)，而在第一站 A 廠的總加工時間為  $TP_{a1}=5+3=8$ ，B 廠的總加工時間為  $TP_{b1}=2+3+3=8$ 。

步驟二： $M=2$  代入公式(1)，得到  $Min\left|\sum_{m=1}^2 TP_{am}-\sum_{m=1}^2 TP_{bm}\right|=Min|(TP_{a1}+TP_{a2})-(TP_{b1}+TP_{b2})|=$

$$Min\left[\left(TP_{a1}+\sum_{j=1}^{c_2} P_{am(j)}\right)-\left(TP_{b1}+\sum_{j=c_2+1}^n P_{bm(j)}\right)\right]=Min\left[\left(8+\sum_{j=1}^{c_2} P_{am(j)}\right)-\left(8+\sum_{j=c_2+1}^n P_{bm(j)}\right)\right]，求$$

出第二站的分割點  $c_2=3$  時，可得到最小值  $=\left[8+(4+3+2)\right]-\left[8+(3+5)\right]=1$ ，因此可以得知在第二站的廠分割點為第三個順位工件之後，亦即第三個順位之前的工件(J1、J4、J3)在 A 廠加工，其餘的工件則在 B 廠加工(J2、J5)，而在第二站 A 廠的總加工時間為  $TP_{a2}=4+3+2=9$ ，B 廠的總加工時間為  $TP_{b2}=3+5=8$ 。

步驟三： $M=3$  代入公式(1)，得到  $Min\left|\sum_{m=1}^3 TP_{am}-\sum_{m=1}^3 TP_{bm}\right|=Min\left[\left(TP_{a1}+TP_{a2}+\sum_{j=1}^{c_3} P_{am(j)}\right)-\left(TP_{b1}+TP_{b2}+\sum_{j=c_3+1}^n P_{bm(j)}\right)\right]，$

求出第三站的分割點  $c_3=2$  時，可得到最小值  $=\left[8+9+(5+3)\right]-\left[8+8+(2+3+3)\right]=1$ ，因此可以得知在第三站的廠分割點為第二個順位工件之後，亦即第二個順位之前的工件(J1、J4)在 A 廠加工，其餘的工件則在 B 廠加工(J3、J2、J5)。

依據此三站廠分割點切割三段染色體(如圖 3.5)，圖中每一站之工件分割點前代表在 A 廠加工，反之則在 B 廠。而每個工件也隨即知道其加工途程，如圖 3.5，工件 3 (J3) 的加工途程為 B1→A2→B3 (B 廠第一站→A 廠第二站→B 廠第三站)可對照圖 3.6 之實線加工途程。

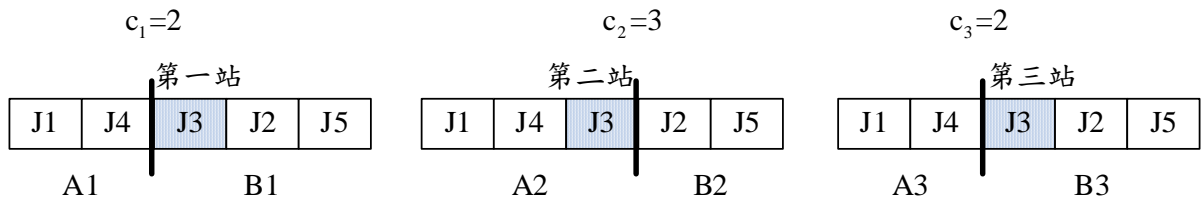


圖 3.5 分廠後染色體解讀

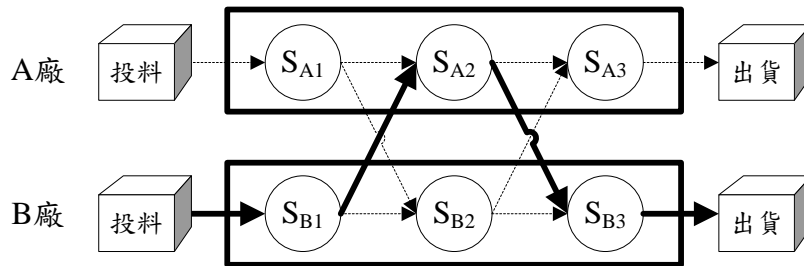


圖 3.6 工件 3 之加工途程

## 2. 加工順序解讀

經由加工途程之染色體解讀後，我們可以知道每個工件的加工途程，但尚未知在各廠各站的工件加工順序，此小節根據本研究的目標：最小化寬裕時間之變異係數 ( $Min CV_s$ )，希望能夠同時兼顧節省整備時間及交期的限制，提出群組演算法，使排程穩健度更高，降低重排程的頻率。

### (1) 群組演算法(GA-EDD-Group)

群組演算法的概念為利用貪婪法(Greedy Method)，試著合併相同工件族之工件，意即一次能只試著合併兩個相同工件族之工件，看合併結果是否較佳來決定是否合併，直到所有工件皆考慮過合併為止。如此即能同時兼顧節省整備時間及交期的限制，使兩者之間取得平衡。

以下演算步驟之步驟一的主要目的是根據交期限制，初步排序加工順序，我們利用 EDD 排序(Early Due Date)來初步決定每廠每站的加工順序，亦即交期越早之工件排至越前面加工。步驟二～步驟五是依據步驟一的結果細部調整加工順序，主要目的是利用貪

算法合併相同工件族之工件，以減少整備時間，並同時能兼顧交期限制。以下則做詳細的介紹。

符號解釋：

$t$ ：第  $t$  次合併

$S_t$ ：第  $t$  次合併之  $CV_s$

$S^*$ ：整體最佳解

演算步驟：

步驟一：各廠各站分別依據 EDD 排序，並代入適合度函數計算出  $CV_s = S_{t=0}$  ( $S^* = S_{t=0}$ )，到步驟二。

步驟二：以第一站的排序為主，依排序考慮各工件，並往前尋找同廠且相同、相近但不相鄰之工件族工件往前合併，到步驟三。若每個工件已考慮過合併，則到步驟五。

步驟三：第二站及第三站在同一廠找到與第一站合併的工件相同之工件則往前合併，若某一站(第二站或第三站)沒有同時找到此兩個工件則此站可省略合併，到步驟四。

步驟四：將三站皆合併好的排序，代入適合度函數計算出  $CV_s = S_{t+1}$  (望小)，若  $S_{t+1} < S^*$ ，則  $S^*$  被  $S_{t+1}$  取代 ( $S^* = S_{t+1}$ )，並利用此次合併的排序繼續考慮下個工件的合併，回到步驟二；反之，若  $S_{t+1} \geq S^*$ ，則不合併，並利用此次合併前的排序繼續考慮下個工件的合併，回到步驟二。

步驟五：停止。最後的排序即為經由群組演算法的最佳排序，而  $S^*$  則為經由群組演算法後的  $CV_s$ 。

釋例：

假設一條染色體經由加工途程解讀後之染色體排序，形成如圖 3.7，圖中  $A_m$  代表 A



廠第  $m$  站的工件排序( $m=1, 2, 3$ )； $B_m$  代表 B 廠第  $m$  站的工件排序( $m=1, 2, 3$ )，表 3.2 為各工件交期及工件族之相關資訊( $J_i$ ：工件， $D_i$ ：交期， $F_i$ ：工件族)。

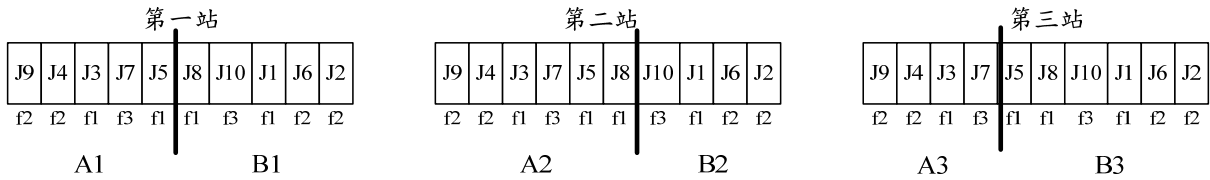


圖 3.7 染色體經加工途程解讀後的染色體排序-(1)

表 3.2 各工件交期及工件族之相關資訊-(1)

$J_i$	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
$D_i$	2	2	3	4	5	6	6	7	8	9
$F_i$	f1	f2	f1	f2	f1	f2	f3	f1	f2	f3

步驟一：各廠各站分別依據 EDD 排序(圖 3.8)，並代入適合度函數計算出  $CV_s$  (假設算出  $S_{t=0}=3$ ，而  $S^*=3$ )，到步驟二。

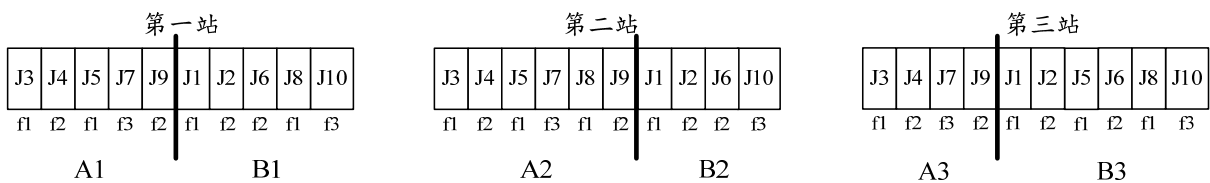


圖 3.8 EDD 排序後的染色體解讀-(1)

步驟二：以第一站的排序為主，依排序考慮各工件(J3、J4、J5...)，考慮到 J5 時，往前尋找到同廠(A 廠)且相同、相近但不相鄰之工件族工件為 J3，並往前合併(如圖 3.9)，到步驟三。

步驟三：第二站在同一廠(A 廠)找到與第一站合併的工件相同之工件(J5、J3)則往前合併，第三站在同一廠(A 廠)沒有同時找到此兩個工件(J5 沒有找到)則此站可省略

合併(如圖 3.9)，到步驟四。

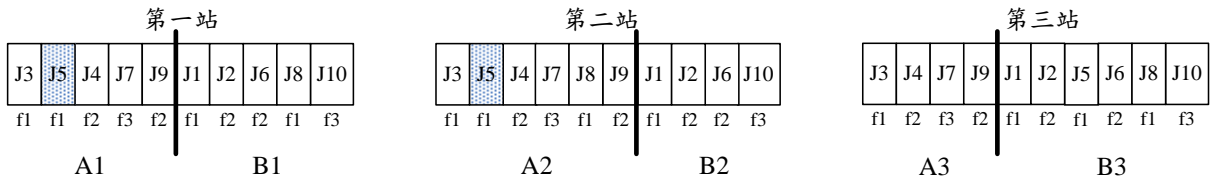


圖 3.9 J5 往前與 J3 合併之染色體解讀

步驟四：將三站皆合併好的排序(圖 3.9)，代入適合度函數計算出  $CV_s$  (假設算出

$S_{t=1}=3.1$ )，由於  $S_{t=1} > S^*$ ，則不合併，並利用此次合併前的排序(圖 3.8)，繼續考慮下個工件的合併，回到步驟二。

步驟二：以第一站的排序為主，依排序考慮下個工件(J7、J9...)，考慮到 J9 時，往前尋找到同廠(A 廠)且相同、相近但不相鄰之工件族工件為 J4，並往前合併(如圖 3.10)，到步驟三。

步驟三：第二站及第三站在同一廠(A 廠)找到與第一站合併的工件相同之工件(J9、J4)並往前合併(如圖 3.10)，到步驟四。

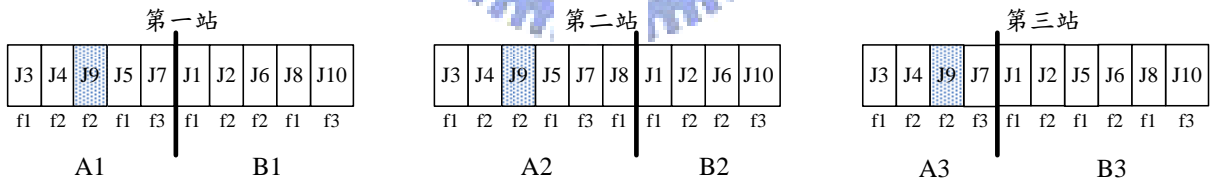


圖 3.10 J9 往前與 J4 合併之染色體解讀

步驟四：將三站皆合併好的排序(圖 3.10)，代入適合度函數計算出  $CV_s$  (假設算出

$S_{t=2}=2.8$ )，由於  $S_{t=2} < S^*$ ， $S^*$  被  $S_{t=2}$  取代( $S^*=S_{t=2}$ )，且利用此次合併的排序(圖 3.10)，繼續考慮下個工件的合併，回到步驟二。

步驟二：以第一站的排序為主，依排序考慮下個工件(J1、J2、J6、J8...)，考慮到 J8 時，往前尋找到同廠(B 廠)且相同、相近但不相鄰之工件族工件為 J1，並往前合併(如圖 3.11)，到步驟三。

步驟三：第二站在同一廠(B 廠)沒有同時找到此兩個工件(J8 沒有找到)則此站可省略合併，第三站在同一廠(B 廠)找到與第一站合併的工件相同之工件(J8、J1)則往前合併(如圖 3.11)，到步驟四。

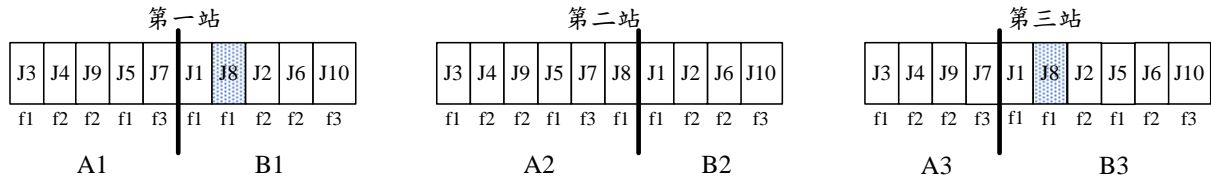


圖 3.11 J8 往前與 J1 合併之染色體解讀

步驟四：將三站皆合併好的排序(圖 3.11)，代入適合度函數計算出  $CV_s$  (假設算出

$S_{t=1}=3.2$ )，由於  $S_{t=3} > S^*$ ，則不合併，並利用此次合併前的排序(圖 3.10)，繼續考慮下個工件的合併，回到步驟二。

步驟二：以第一站的排序為主，依排序考慮下個工件(J10)，往前沒有尋找到相同、相近但不相鄰之工件族工件，且為最後一個考慮合併的工件，代表所有工件已考慮過合併，則到步驟五。

步驟五：停止合併。最後的排序(圖 3.10)即為經由群組演算法的最佳排序，而  $S^*=2.8$  則為其  $CV_s$ 。

## (2) 家族式派工(GA-EDD-Family)

家族式派工是一種可以大量節省整備時間的派工方法，意即強制將相同工件族的工件排序一起加工，但本研究因應顧客的需求有不同交期的限制，這種家族式派工法只考慮到節省大量的整備時間，但卻沒有考慮到交期的限制而強制將相同工件族排在一起加工，並不適合本研究的研究目標，我們將於第四章與群組演算法做結果比較，以下則先介紹家族式派工法的演算步驟。

演算步驟：

步驟一：各廠各站分別依據 EDD 排序，到步驟二。

步驟二：決定工件族間(Among Family)的排序，各廠各站分別將每個工件族第一次出現順序的先後做為工件族間的排序，到步驟三。

步驟三：各廠各站分別依據工件族間的序，依序將相同工件族的工件排入，來決定各工件族內(Within Family)的順序，此排序即為家族式派工後的最佳排序。

釋例：

假設一條染色體經由加工途程解讀後之染色體排序，形成如圖 3. 12，圖中  $A_m$  代表 A 廠第  $m$  站的工件排序( $m=1, 2, 3$ )； $B_m$  代表 B 廠第  $m$  站的工件排序( $m=1, 2, 3$ )，表 3. 3 為各工件交期及工件族之相關資訊( $J_i$ ：工件， $D_i$ ：交期， $F_i$ ：工件族)。

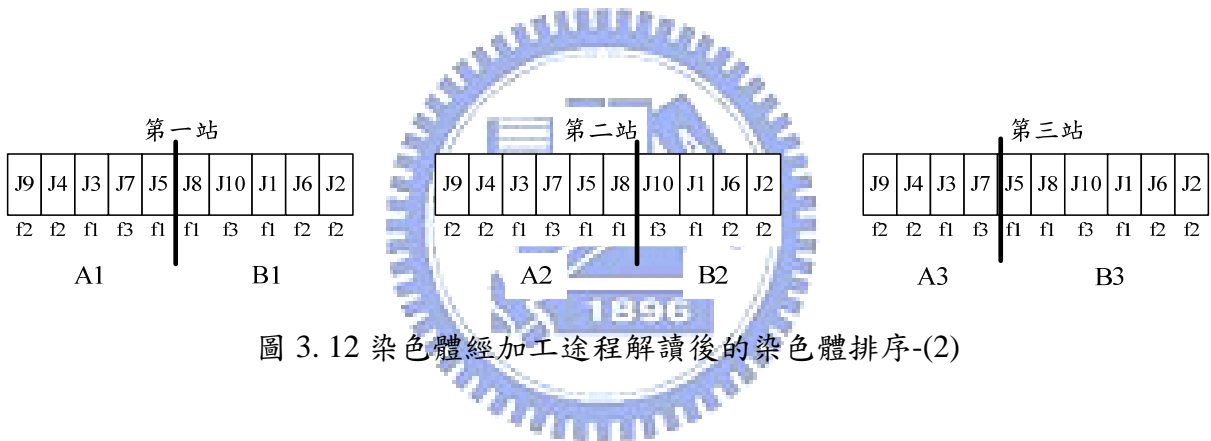


圖 3. 12 染色體經加工途程解讀後的染色體排序-(2)

表 3. 3 各工件交期及工件族之相關資訊-(2)

$J_i$	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10
$D_i$	2	2	3	4	5	6	6	7	8	9
$F_i$	f1	f2	f1	f2	f1	f2	f3	f1	f2	f3

步驟一：各廠各站分別依據 EDD 排序(如圖 3. 13)，到步驟二。

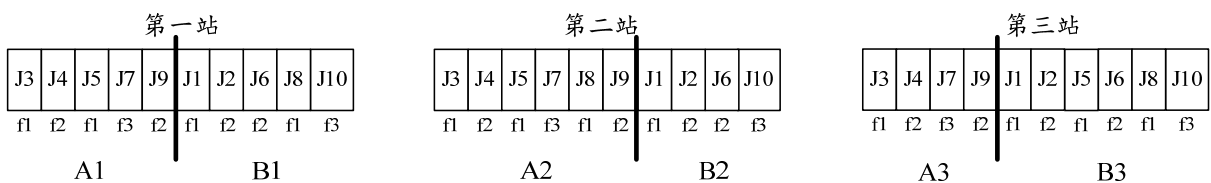


圖 3.13 EDD 排序後的染色體解讀-(2)

步驟二：各廠各站分別將每個工件族第一次出現順序的先後做為工件族間的排序，如 A1 第一次出現的工件族為 f1，接著為 f2、f3，故工件族間的排序為 f1→f2→f3，並以此類推其他各廠各站(如圖 3.14)，到步驟三。

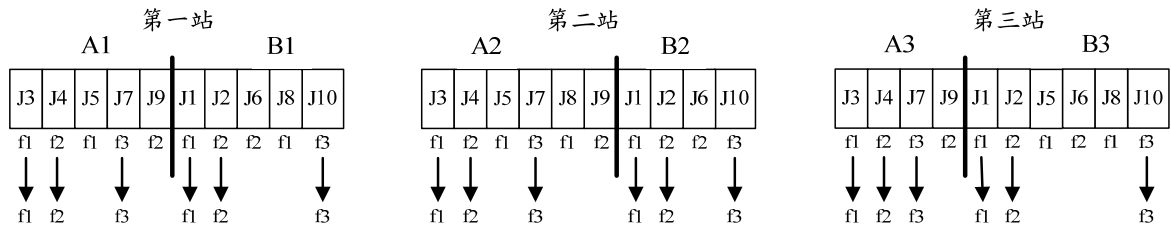


圖 3.14 工件族間的排序

步驟三：各廠各站分別依據工件族間的序，依序將相同工件族的工件排入，來決定各工件族內(Within Family)的順序，如 A1 工件族間的序為 f1→f2→f3(圖 3.14)，A1 中 f1 的工件依序有 J3、J5；f2 的工件依序有 J4、J9；f3 的工件有 J7，故 A1 依據家族式派工後的順序為 J3→J5→J4→J9→J7，以此類推其他各廠各站，最後的排序即為家族式派工後的最佳排序(如圖 3.15)。

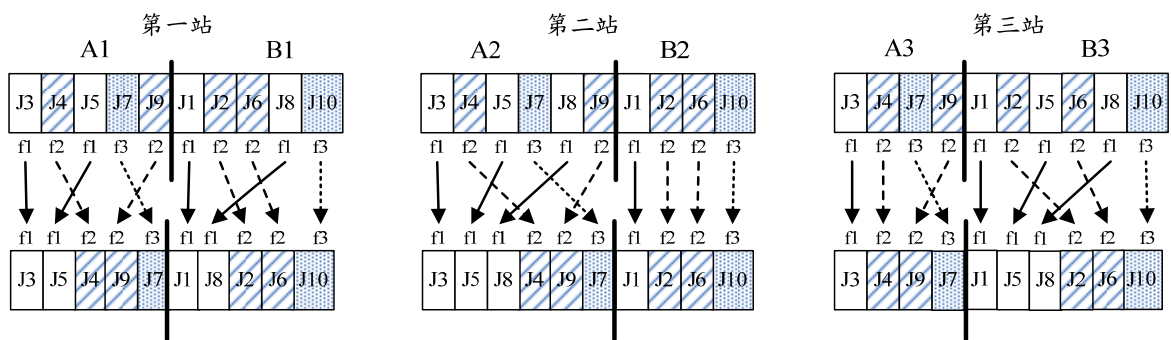


圖 3.15 家族式派工後的最佳排序

### 3.2.3 適合度函數

經由上一節之染色體解讀後，我們將解讀後的染色體代入適合度函數，以求得染色

體解讀後的排程之目標值( $CV_s$ )，本研究的目標值為最小化寬裕時間之變異係數( $Min CV_s$ )，以下介紹適合度函數在基因演算法中的計算方式。

符號解釋：

$n$ ：總工件數

$pm(j)$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件。

$n_{pm}$ ：在工廠  $p$  的第  $m$  個機台之總加工工件數。

$(j)$ ：加工順序為  $j$  的工件， $j=1, 2, \dots, N_{pm}$ 。

$p$ ：工廠  $p$ ， $p=1, 2$  (1 代表工廠 A，2 代表工廠 B)。

$m$ ：機台  $m$ ， $m=1, 2, 3$ 。

$d_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之交期。

$AT_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之到達時間。

$ST_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之開始加工時間。

$PT_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之加工時間。

$CT_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之完成時間。

$CT_{p',m-1,pm(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件在第  $m-1$  個機台之完成時間，

而  $p'$  代表在第  $m-1$  個機台加工之工廠非 A 即 B。

$f_{p,m(j)}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件之所屬工件族。

$S_{f_{p,m(j)}}$ ：在工廠  $p$  的第  $m$  個機台，加工順序為  $j$  的工件所屬工件族之整備時間。

$TS$ ：跨廠運輸時間。

適合度函數計算：

以下公式(2)、(3)為計算各工件在各廠各站之到達時間；公式(4)、(5)、(6)、(7)為計

算各工件在各廠各站之開始加工時間；公式(8)則為計算各工件在各廠各站之完成時間；公式(9)為計算各工件的寬裕時間；而公式(10)即為本研究之目標式。以下則一一介紹各公式之使用情況。

$$AT_{p,m,(j)}=0 \quad \text{for } j=1, 2, \dots, n_{pm}, m=1, p=1, 2 \quad (2)$$

$$AT_{p,m,(j)}=CT_{p',m-1,pm(j)}+|p-p'|\times TS \quad \text{for } j=1, 2, \dots, n_{pm}, m=2, 3, p=1, 2 \quad (3)$$

公式(2)為每個工件在第一站時，到達的時間皆為0。

公式(3)計算每個工件到達第二或第三站的時間為前一站的完成時間加上運輸時間，若  $p-p'=0$  代表沒有跨廠，運輸時間為零，反之  $p-p'=1$  則需要運輸時間。

$$ST_{p,m,(j)}=AT_{p,m,(j)}+S_{f_{p,m,(j)}} \quad \text{for } j=1, m=1, 2, 3, p=1, 2 \quad (4)$$

公式(4)計算每廠每站排序為第一順位之加工工件皆須整備時間，因此第一順位之加工工件的開始加工時間為到達時間加上整備時間。

$$ST_{p,m,(j)}=CT_{p,m,(j-1)}+Min\left\{\left|f_{p,m,(j)}-f_{p,m,(j-1)}\right|, I\right\}\times S_{f_{p,m,(j)}} \quad \text{for } j=2, 3, \dots, n_{pm}, m=1, p=1, 2 \quad (5)$$

公式(5)計算每廠的第一站非第一順位之加工工件的開始加工時間，即上一加工順位工件之完成時間加上是否必須整備時間，整備時間與否視與上一加工順位工件之工件族是否相同而定， $Min\left\{\left|f_{p,m,(j)}-f_{p,m,(j-1)}\right|, I\right\}=0$ ，即為與上一加工順位工件之工件族相同，因此整備時間為零；反之， $Min\left\{\left|f_{p,m,(j)}-f_{p,m,(j-1)}\right|, I\right\}=1$ ，則為不相同，故需要整備時間。

$$ST_{p,m,(j)}=CT_{p,m,(j-1)}+Min\left\{\left|f_{p,m,(j)}-f_{p,m,(j-1)}\right|, I\right\}\times S_{f_{p,m,(j)}} \\ \text{if } AT_{p,m,(j)}<CT_{p,m,(j-1)}+Min\left\{\left|f_{p,m,(j)}-f_{p,m,(j-1)}\right|, I\right\}\times S_{f_{p,m,(j)}} \quad (6) \\ \text{for } j=2, 3, \dots, n_{pm}, m=2, 3, p=1, 2$$

公式(6)與公式(5)相同，但使用情況不同，使用此公式的情況為：每廠的第二或第三站非第一順位之加工工件，當上一順位工件之完成時間加上整備時間晚於工件到達時間(如圖 3.16、圖 3.17)，亦即工件到達之前來不及在上一順位工件完成後整備完成，此時工件的可開始加工時間為上一順位工件之完成時間加上整備時間。

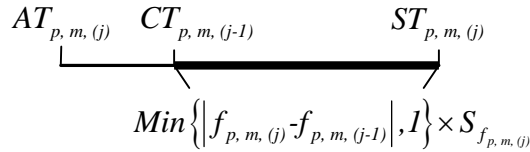


圖 3.16 上一順位工件之完成時間加上整備時間晚於工件到達時間-(1)

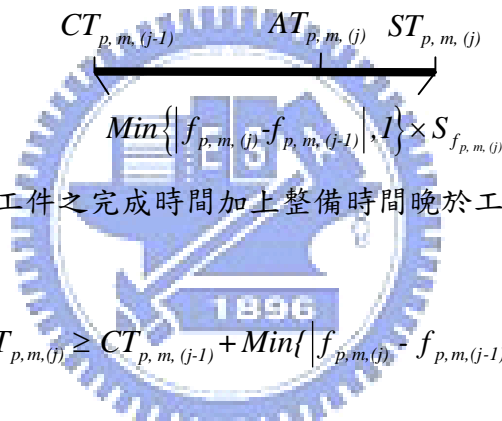


圖 3.17 上一順位工件之完成時間加上整備時間晚於工件到達時間-(2)

$$ST_{p,m,(j)} = AT_{p,m,(j)} \quad \text{if } AT_{p,m,(j)} \geq CT_{p,m,(j-1)} + \text{Min}\{|f_{p,m,(j)} - f_{p,m,(j-1)}|, I\} \times S_{f_{p,m,(j)}} \quad (7)$$

for  $j=2, 3, \dots, n_{pm}, m=2, 3, p=1, 2$

公式(7)亦為工件開始加工時間，使用此公式的情況為：每廠的第二或第三站非第一順位之加工工件，上一順位工件之完成時間加上整備時間早於工件到達時間，亦即在工件到達之前早就可以在上一順位工件完成後整備完成，故工件的開始加工時間即為工件之到達時間(如圖 3.18)。

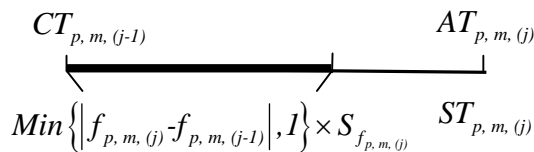


圖 3.18 上一順位工件之完成時間加上整備時間早於工件到達時間



$$CT_{p,m,(j)} = ST_{p,m,(j)} + PT_{p,m,(j)} \quad \text{for } j=1, 2, \dots, n_{pm}, m=1, 2, 3, p=1, 2 \quad (8)$$

公式(8)計算每廠在第一、二及第三站之工件的完成時間，為工件開始加工時間加上工件之加工時間。

$$S_{p,m,(j)} = d_{p,m,(j)} - CT_{p,m,(j)} \quad \text{for } j=1, 2, \dots, n_{pm}, p=1, 2, m=3 \quad (9)$$

公式(9)計算各工件的寬裕時間，為各工件的交期減去各工件的最後完工時間(即每個工件在第三站之完成時間)。

$$\text{Min} \left( \frac{\sigma_s}{\bar{X}_s} \right), \text{ 其中 } \left( \bar{X}_s = \frac{\sum_{p=1}^2 \sum_{j=1}^{n_{pm}} S_{p,m,(j)}}{n}, \sigma_s = \sqrt{\frac{\sum_{p=1}^2 \sum_{j=1}^{n_{pm}} (S_{p,m,(j)} - \bar{X}_s)^2}{n-1}} \right) \quad (10)$$

st.  $\bar{X}_s \geq 0$  for  $j=1, 2, \dots, n_{pm}, p=1, 2, m=3$

公式(10)即為本研究之目標式，目標使寬裕時間之變異係數能夠最小化 ( $\text{Min } CV_s, CV_s = \frac{\sigma_s}{\bar{X}_s}$ )，此目標值為望小，所以必須限制寬裕時間平均為正數( $\bar{X}_s \geq 0$ )，

否則目標值雖然很小，但寬裕時間平均卻也很小( $\bar{X}_s \leq 0$ )，違反了本研究的目的。

### 3.2.4 染色體交配、突變

#### 1. 交配(Crossover)

基因演算法中交配之演化方式，每一次交配，會從母代中挑選兩個染色體，經由交配後，會產生新的兩個染色體，稱為子代。假設所設定的交配率為 $\alpha$ ，而母代數量為 $N$ 個，所以經由交配後產生新的染色體子代數量為 $\alpha \cdot N$ 個。本研究基因演算法中使用線性順序交配(LOX: Linear Order Crossover Operator)之遺傳法則，如圖 3.19，可分為兩步驟操作：

步驟一：從母代中挑選兩個染色體(母代一、母代二)，隨機選取兩個點，依據此兩個隨

機切點，將此兩個染色體分為前段、中段與後段，子代一複製母代一之中段部份；子代二複製母代二之中段部份。

步驟二：將母代二與子代一尚未重複的編碼依序排入子代一之前段與後段尚未填滿之基因格；將母代一與子代二尚未重複的編碼依序排入子代二之前段與後段尚未填滿之基因格。

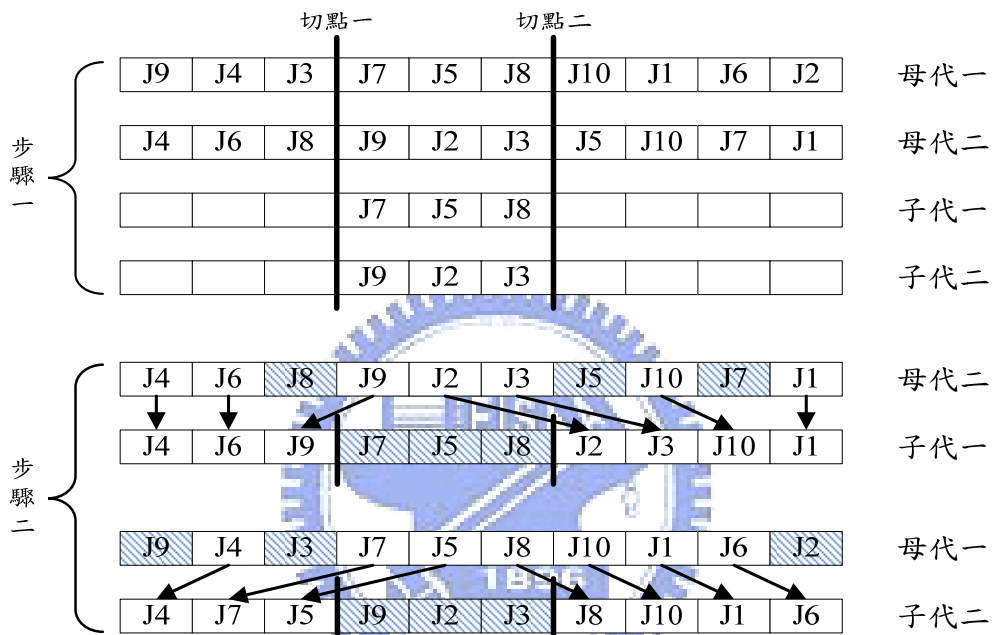


圖 3. 19 線性順序交配(LOX)

## 2. 突變(Mutation)

每一次的突變，會從母代中挑選一個染色體，經由突變後，會產生新的一個染色體，突變通常會與交配互相搭配使用，可以避免太早陷入局部最佳解。而本研究使用 Swap 的突變方式，如圖 3. 20 所示，從母代中挑選出一條染色體，並在此條染色體中隨機選取兩個不同的基因格，將其對調產生一條新的染色體子代。

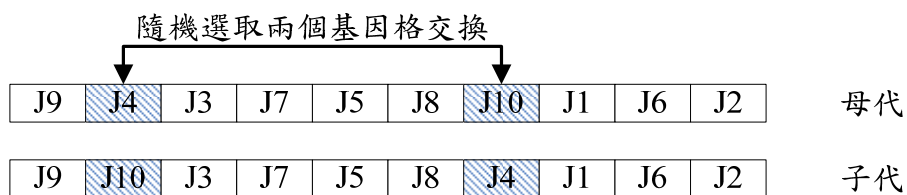


圖 3. 20 突變方式(Swap)

### 3.2.5 染色體複製

經由子代交配、突變演化的運算後產生新的子代染色體，與原先母代的染色體加總共有  $Q$  個染色體，必須從  $Q$  個染色體中選出  $Q/2$  個染色體當做下一代之母代。挑選的方法是將  $Q$  個染色體以適應函數值的優劣，由優到劣排序。最優的適應函數值必須要保留到下一代，其餘的染色體則以俄羅斯輪盤法(Roulette wheel)選取(如圖 3.21)，輪盤上所佔面積越大存活的機率越高，反之則越小。圖中代表交配池中有七條染色體，為依據這些染色體的適合度函數值所組成的輪盤，被挑選中的染色體則可被複製到下一代成為新的母代，而未被選中者則被淘汰。

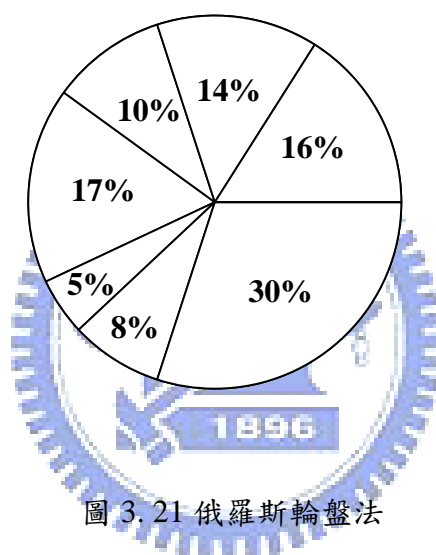


圖 3.21 俄羅斯輪盤法

### 3.2.6 終止條件

本研究使用兩種基因演算之終止條件，亦即達到以下兩種條件之其中一種即終止：

1. 達到所設定的最大演化代數上限，意即設定最大演化代數為  $X$  代，基因演化到達  $X$  代即結束。
2. 達到所設定的最佳解維持代數，意即設定最佳解維持代數為  $Y$  代，基因演化最佳解連續維持代  $Y$  代沒有改變即結束。

第三章是以基因演算法為架構做研究方法的介紹，而基因演算法的執行步驟已於第二章詳述，在此不多做介紹。接著，我們將於第四章做實驗，驗證研究方法的優劣。

## 第四章 實驗分析與結果

本論文提出群組演算法來求解雙流線型工廠的排程，此演算法兼顧考慮工件族整備時間及交期限限制所發展出的演算法，為了與家族式派工比較演算法的績效，我們執行了在各種不同情境下的實驗。以下資料為實驗情境的假設、基因演算法參數設定及求解品質衡量指標：

### 1. 實驗情境：

- a. 工件數：20、40、60、80。
- b. 工件族個數：2、4、6、8。
- c. 加工時間：各工件在各站的加工時間皆為均勻分配  $U(6\sim 26)$ /小時。
- d. A、B 兩廠加工時間效率比：一比一。
- e. 交期：20 個工件— $U(250\sim 400)$  /小時；40 個工件— $U(250\sim 550)$  /小時；60 個工件— $U(250\sim 750)$  /小時；80 個工件— $U(250\sim 1000)$  /小時。
- f. 整備時間：各工件族在各廠各站的整備時間皆相同，設定三種整備時間情境分別為平均加工時間的 5%、15%、30% (0.8 小時、2.4 小時、4.8 小時)。
- g. 運輸時間：8 小時。

### 2. 基因演算法的參數設定：

- a. 初始母代染色體數 100 條。
- b. 子代數量=(母代數量×交配率)+(母代數量×突變率)
- c. 交配率：0.8。
- d. 突變率：0.2。
- e. 演算停止條件：最佳解代數維持 1,000 世代；執行代數達到 100,000 世代。

### 3. 求解品質衡量指標(改善率)：

- a. 群組演算法(GA-EDD-Group)改善率( $R_g$ )： $R_g = \frac{CV_i - CV_g}{CV_i}$ ，其中  $CV_i$  是只有解讀至各廠各站僅有以 EDD 排序之加工順序(因為沒有考慮相同工件族合併，在此稱為 Individul-based)； $CV_g$  則是群組演算法之績效。

- b. 家族式派工(GA-EDD-Family)改善率( $R_f$ ):  $R_f = \frac{CV_i - CV_f}{CV_i}$ ，其中  $CV_f$  是家族式派工之績效。

#### 4.1 求解品質

實驗結果如表 4. 1，我們將此些數據以  $CV_i$  為基礎計算出衡量指標(改善率)如表 4. 2，我們發現家族式派工的改善率幾乎皆為負數，代表家族式派工的結果比僅有 EDD 排序的結果還差，更證明了我們先前的假設，在各工件皆有不同的交期情況下，家族式派工法只考慮到節省大量的整備時間，但卻沒有考慮到交期的限制而強制將相同工件族排在一起加工，雖然可以節省很多整備時間，但卻容易造成交期晚的工件為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，反而讓績效指標變差，並不適合本研究的研究目標。

相較於群組演算法的改善率皆為正數，代表群組演算法的結果比僅以 EDD 排序的結果還佳，因為群組演算法的特色是以 EDD 排序為基礎，不斷的利用貪婪法試著合併相同工件族的工件，績效有變好才合併，反之則不合併，如此即可同時考慮到節省整備時間及交期限制下做績效的改善，因此我們建議在求解雙流線型工廠，兼顧考慮節省工件族整備時間及各工件皆有不同交期限制的情境之下，利用群組演算法可以求得近似最佳解排程。

接著，我們分析群組演算法改善率的趨勢，首先分析各工件族個數不同的情境下，各工件的改善率趨勢，如圖 4. 1，圖中顯示，無論在哪一種工件數的情況下，當工件族個數越大時，其改善率跟著增加，但當工件族大到某種程度時，其改善率則又呈下降的趨勢。分析其原因為當工件族個數很大時，會使相同工件族的工件排序相隔很遠，其工件的交期亦差距很大，導致不易合併的情況，故其改善率則呈現下降的趨勢。反之，當工件族個數很小時，如僅以 EDD 排序，就已經很容易使得相同工件族的工件排序在一

起，如此若使用群組演算法再做合併的效益並不高，故其改善率也較低。

然而，分析各整備時間不同的情境下，各工件的改善率趨勢，如圖 4.2，無論何種工件數，其改善率的趨勢隨著整備時間越大，改善率則越大。原因是因為當整備時間越大時，使用群組演算法所能節省的整備時間也越大，進而使改善率也隨之增大。

根據以上實驗數據分析的結果，我們得到的結論是無論在何種情境之下，群組演算法皆有一定的績效，尤其在整備時間大以及工件族個數大，但工件族個數又不能過大的情境之下，群組演算法會有很好的績效。

表 4.1 三種演算法各情境之實驗數據

Job 數	Setup time(hrs)	Family 數			Family 數			Family 數			Family 數		
		目標值			目標值			目標值			目標值		
		$CV_i$	$CV_f$	$CV_g$	$CV_i$	$CV_f$	$CV_g$	$CV_i$	$CV_f$	$CV_g$	$CV_i$	$CV_f$	$CV_g$
20	0.8	0.0925	0.1129	0.0916	0.0960	0.1109	0.0946	0.0972	0.1020	0.0958	0.0964	0.0979	0.0961
	2.4	0.1015	0.1171	0.1003	0.1117	0.1210	0.1094	0.1194	0.1187	0.1172	0.1211	0.1222	0.1207
	4.8	0.1119	0.1206	0.1106	0.1376	0.1382	0.1310	0.1553	0.1468	0.1456	0.1633	0.1634	0.1619
40	0.8	0.0968	0.1321	0.0931	0.1009	0.2150	0.0943	0.1064	0.1757	0.1004	0.1071	0.1375	0.1060
	2.4	0.1066	0.1358	0.1019	0.1266	0.2366	0.1172	0.1476	0.1880	0.1309	0.1506	0.1745	0.1413
	4.8	0.1194	0.1414	0.1092	0.1654	0.2501	0.1461	0.2170	0.2179	0.1792	0.2363	0.2446	0.2138
60	0.8	0.1079	0.1798	0.1066	0.1124	0.3894	0.1121	0.1140	0.3700	0.1123	0.1188	0.2961	0.1146
	2.4	0.1234	0.1713	0.1181	0.1406	0.4192	0.1341	0.1569	0.3966	0.1439	0.1657	0.3214	0.1571
	4.8	0.1386	0.1899	0.1279	0.1987	0.4558	0.1746	0.2509	0.4645	0.2085	0.3158	0.4151	0.2757
80	0.8	0.0848	0.2083	0.0818	0.0939	0.4774	0.0888	0.0966	0.5185	0.0920	0.0990	0.4720	0.0958
	2.4	0.1001	0.2152	0.0959	0.1290	0.4997	0.1219	0.1579	0.5329	0.1428	0.1737	0.5382	0.1536
	4.8	0.1197	0.2185	0.1140	0.2178	0.5448	0.1880	0.3349	0.6044	0.2514	0.4235	0.6450	0.3283

表 4.2 家族式派工與群組演算法各情境之改善率

Job 數	Setup time(hrs)	Family 數		4		8		16			
		改善率 (%)		$R_f$ (%)	$R_g$ (%)	$R_f$ (%)	$R_g$ (%)	$R_f$ (%)	$R_g$ (%)		
20	0.8			-22.04	0.96	-15.43	1.49	-4.95	1.40	-1.53	0.25
	2.4			-15.40	1.10	-8.28	2.08	0.54	1.85	-0.90	0.36
	4.8			-7.74	1.16	-0.43	4.76	5.44	6.25	-0.02	0.86
40	0.8			-36.55	3.79	-113.05	6.58	-65.25	5.61	-28.35	1.03
	2.4			-27.33	4.44	-86.89	7.39	-27.37	11.33	-15.86	6.20
	4.8			-18.44	8.56	-51.20	11.66	-0.44	17.40	-3.53	9.51
60	0.8			-66.70	1.14	-246.60	0.25	-224.62	1.46	-149.17	3.55
	2.4			-38.80	4.30	-198.23	4.61	-152.75	8.32	-93.94	5.19
	4.8			-37.04	7.67	-129.41	12.15	-85.14	16.89	-31.43	12.72
80	0.8			-145.66	3.58	-408.21	5.47	-437.01	4.73	-376.78	3.20
	2.4			-114.97	4.19	-287.22	5.53	-237.52	9.57	-209.87	11.53
	4.8			-82.50	4.73	-150.12	13.69	-80.46	24.93	-52.32	22.47

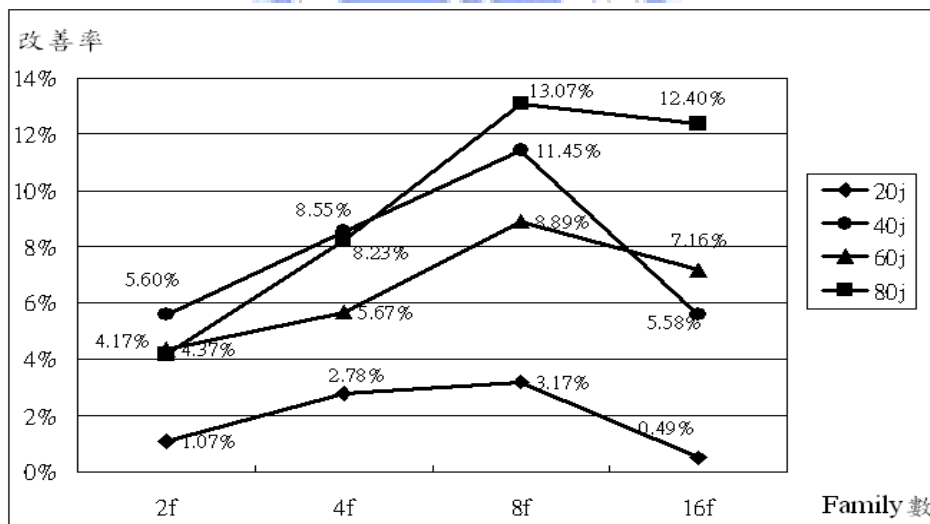


圖 4.1 群組演算法之工件族個數不同的改善率趨勢

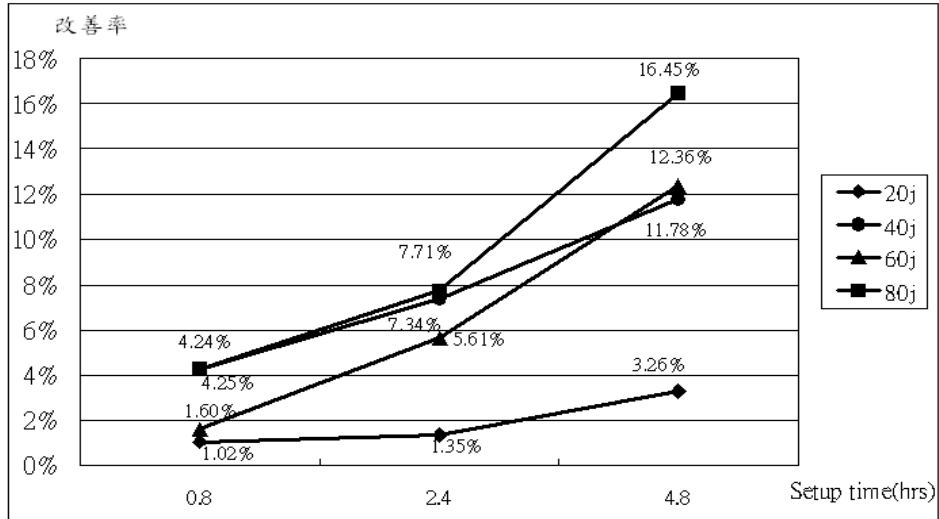


圖 4.2 群組演算法之整備時間不同的改善率趨勢

#### 4.1 求解速度

我們分別將三種演算法(僅以 EDD 排序、家族式派工、群組演算法)之各工件數的求解速度取其平均，比較各種演算法的求解速度(圖 4.3)，求解速度為僅以 EDD 排序 (GA-EDD-Individual)最快，其次為家族式派工(GA-EDD-Family)，最後是群組演算法 (GA-EDD-Group)的求解時間最慢。群組演算法的求解速度慢的原因是因為它不斷的利用貪婪法求其近似最佳解，尤其在 80 個工件時，需要 4.2 個多小時，雖然其求解時間比較長，但是在實務上，提前排程是可被接受的。

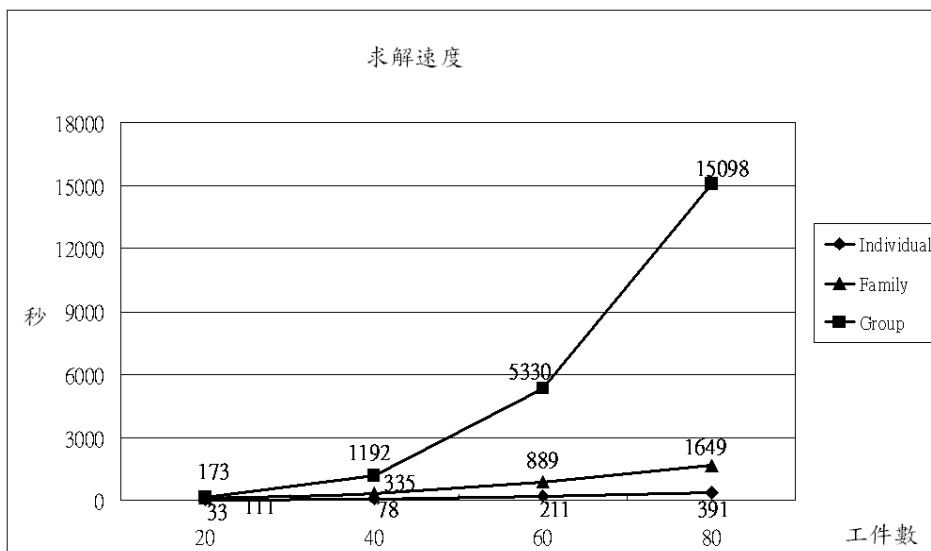


圖 4.3 三種演算法之求解速度



## 第五章 結論與未來研究

本論文探討一個雙流線型工廠的問題，此問題為兩座工廠皆為多階段流線型的生產系統，兩廠間同一站有相同加工製程機台，可允許跨廠支援加工，並考慮工件族整備時間以及有交期的限制下，建立一個穩健的生產排程。我們必須兼顧考慮減少整備時間以及交期限制的情況下合併相同工件族的工件，在求得近似最佳績效之下能夠決定各工件的跨廠加工途程及各機台工件加工順序。

本研究提出了群組演算法，同時考慮合併相同工件族之工件以減少整備時間，以及各工件有不同的交期限制，無法將相同工件族之工件全部強制合併，否則則容易造成交期晚的工件為了節省整備時間而排序在前面生產；交期早的工件則排序在後面生產，造成績效變差。基於以上理由，發展出群組演算法，並證實群組演算法(GA-EDD-Group)有良好的績效。尤其在整備時間大以及工件族個數大，但工件族個數又不能太極端(過大)的情境之下，群組演算法會有很好的績效。

群組演算法的是以 EDD 排序為基礎合併相同工件族工件，合併績效好則選擇合併，反之則不合併，所以群組演算法退而至少比僅以 EDD 排序好，進而可能合併到最後可形成家族式派工的排序，不但節省大量的整備時間而且還考慮了交期的限制。故不論任何的資料設定(Data Set)，群組演算法都有一定的績效，意即在實務上，群組演算法會有一定的穩健度，如此通用性之大，則是本研究的貢獻。

然而，本研究還有一些延伸的空間。本研究之各工件的工件族假設為不會隨著站別不同而改變，未來研究則可以考慮會隨著站別而改變各工件之工件族。而針對 Family 數很大的時候，EDD-Group 的改善率呈下降的狀態，此種情況與 Job 數、Family 數以及 Due date 之間有著很大的相依性，可做當大到何種程度時才會開始呈現下降的分析。另外，可以利用其他巨集演算法(Meta-heuristic)配合群組演算法，求解此雙流線型工廠的

排程與本研究之基因演算法的績效，做求解品質及效率的比較。



## 參考文獻

### 英文部份：

- Allahverdi A., Ng C. T., Cheng T. C. E. and Kovalyov M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Poerational Research* **187**: 985-1032.
- Bonfill A, Espuna A, Puigjaner L. (2008). Decision support framework for coordinated production and transport scheduling in SCM. *Comouters & Chemical Engineering* **32**: 1206-1224.
- Eom D.H., Shin H.J., Kwun I.H., Shim J.K. and Kim S.S. (2002). Scheduling jobs on parallel machines with sequence-dependent family set-up times. *International Journal of Advanced Manufacturing Technology* **19**: 926-932
- Guinet A. (2001). Multi-site planning: A transshipment problem. *International Journal of Production Economics* **74**: 21-32.
- Gnoni M.G., Iavagnilio R., Mossa G., Mummolo G, Di Leva A. (2003). Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. *International Journal of Production Economics* **85**: 251-262.
- Graham, R. L., Lawler, E. L., Lenstra, J. K. and Kan, R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics* **5**: 287–326.
- Holland, J. H. (1975). *Adaptation in Neural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press.
- Jolai F. (2005). Minimizing number of tardy jobs on a batch processing machine with incompatible job families. *European Journal of Operational Research* **162**: 184-190
- Lee Y. H., Jeong C. S and Moon C. (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers & Industrial Engineering* **43**: 351-374.

- Levis A. A. and Papageorgiou L. G. (2004). A hierarchical solution approach for multi-site capacity planning under uncertainty in the pharmaceutical industry. *Computers & Chemical Engineering* **28**: 707-725.
- Moon, C., Kim, J., and Hur, S. (2002). Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain. *Computers & Industrial Engineering* **43**: 331-349.
- Moon C. and Seo Y. (2005). Evolutionary algorithm for advanced process planning and scheduling in a multi-plant. *Computers & Industrial Engineering* **48**: 311-325.
- Omar M.K., Teo S.C. (2006). Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach. *Applied Mathematics and Computation* **181**: 1008-1017.
- Potts C. N. and Kovalyov M. Y. (2000). Scheduling with batching: A review. *European Journal of Operational Research* **120**: 228-249.
- Pearn W.L., Chung S.H., Yang M.H., Shiao K.P. (2008). Solution strategies for multi-stage wafer probing scheduling problem with reentry. *Journal of the Operational Research Society* **59**: 637-651.
- Sambasivan M. and Yahya S. (2005). A Lagrangean-based heuristic for multi-plant, multi-item, multi-period capacitated lot-sizing problems with inter-plant transfers. *Computers & Operations Research* **32**: 537-555.
- Srinoi P., Shayan E., Ghotb F. (2006). A fuzzy logic modelling of dynamic scheduling in FMS. *International Journal of Production Research* **44**: 2183-2203
- Toba, H., Izumi, H., Hatada, H. and Chikushima, T. (2005). Dynamic load balancing among multiple fabrication lines through estimation of minimum inter-operation time. *IEEE Transactions on Semiconductor Manufacturing* **18**: 202-213.
- Vercellis, C. (1999). Multi-plant production planning in capacitated self-configuring two-stage serial systems. *European Journal of Operational Research* **119**: 451-460.

- Wu, M. C. and Chang, W. J. (2007). A short-term capacity trading method for semiconductor fabs with partnership. *Expert Systems with Applications* **33**: 476-483.
- Wu, M. C., Chen, C. F. and Shih, C. F. (2008). Route planning for two wafer fabs with capacity-sharing mechanisms. *International Journal of Production Research*. (DOI: 10.1080/00207540802172029)
- Yuan J.J., Liu Z.H., Ng C.T., Cheng T. C. E. (2006). Single machine batch scheduling problem with family setup times and release dates to minimize makespan. *Journal of scheduling* **9**: 499-513.
- Yimer A.D., Demirli K. (2009). Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines. *International Journal of Approximate Reasoning* **50**: 117-137.



中文部分：

李佳諭，「流程式生產系統下混線生產重排程之研究」，屏東科技大學，碩士論文，民國96年。